

공고특허특0184627

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)(51) Int. Cl. 6
H04N 7/173(45) 공고일자 1999년05월01일
(11) 공고번호 특0184627
(24) 등록일자 1998년12월21일

(21) 출원번호	10-1995-0029351	(65) 공개번호	특1996-0011859
(22) 출원일자	1995년09월07일	(43) 공개일자	1996년04월20일
(30) 우선권주장	302,624 1994년09월08일 미국(US)		
(73) 특허권자	인터넷서널 비즈니스 머신즈 코포레이션 윌리엄 티. 엘리스 미국 뉴욕 10504 아몬크		
(72) 발명자	로렌조 펠콘 2세 미국 캘리포니아 95123 산 호세 오미라 드라이브 253 애쉬호크 라즈 색시나 미국 캘리포니아 95120 산 호세 벨리 퀘일 서클 1236		
(74) 대리인	이병호 최달용		

심사관 : 이노성

(54) 분산형 비디오 데이터 저장 장치를 구비하고 비디오에 최적화된 매체 스트리머

요약

매체 스트리머(10)는 최소한 하나의 비디오 표현에 대한 디지털 표시를 저장하기 위한 대용량 저장 장치를 포함하는, 최소한 하나의 저장 노드(16)를 포함한다. 상기 대용량 저장 장치는 다수의 대용량 저장 장치 유닛들로 구성된다. 최소한 하나의 비디오 표현은 그 전체를 표시하는데 T 만큼의 시간이 요구되며, 다수의 N 개의 데이터 블록들에 저장된다. 각 데이터 블록은 상기 비디오 표현의 약 T/N 기간에 대응하는 데이터를 저장한다. 매체 스트리머는 또한 다수의 통신 노드(14)들을 포함하는데, 각각의 통신 노드는 비디오 표현의 디지털 표시를 받기 위해 최소한 하나의 저장 노드의 출력에 연결된 최소한 하나의 입력 포트를 구비한다. 또한 상기 통신 노드는 상기 디지털 표현의 소비자에게 하나의 데이터 스트림으로 디지털 표시를 전송하는 출력 포트를 다수 포함한다. 디지털 표시의 N 개 데이터 블록은 X 개의 스트라이프들로 분할되며, 여기서 데이터 블록 1, X+1, 2*X+1, ... 등은 X개 스트라이프들중 첫 번째 스트라이프에 연관되며, 데이터 블록 2, X+2, 2*X+2, ... 등은 X개 스트라이프들중 두 번째 스트라이프에 연관되며, 상기 X 개 스트라이프들 각각은 다수의 대용량 저장 장치 유닛들중 서로 다른 하나에 저장된다. 다수의 대용량 저장 장치 유닛은 바람직하게는 비디오 표현의 디지털 표시에 대한 하나의 사본을 저장한다. 상기 X 개 스트라이프들은 다수의 데이터 스트림들이 N 개 데이터 블록들중 동일한 블록을 동시에 전달할 수 있게 하는 방식으로 판독되거나, 다수의 데이터 스트림들이 N 개 데이터 블록들중 서로 다른 블록을 동시에 전달할 수 있게 하는 방식으로 판독된다.

명세서

[발명의 명칭]

분산형 비디오 데이터 저장 장치를 구비하고 비디오에 최적화된 매체 스트리머

[도면의 간단한 설명]

제1도는 본 발명을 구체화하는 매체 스트리머의 블록도.

제1a도는 제1도에 보인 회전 스위치의 세부을 나타내는 블록도.

제1b도는 제1도에 보인 테이프 저장 노드의 세부을 나타내는 블록도.

제1c도는 제1도에 보인 디스크 저장 노드의 세부을 나타내는 블록도.

제1d도는 제1도에 보인 통신 노드의 세부을 나타내는 블록도.

제2도는 높은 우선순위로 실행되는 비디오 스트림 출력 제어 명령들의 목록과, 낮은 우선순위로 실행되는 데이터 관리 명령들의 목록.

제3도는 통신 노드의 데이터 흐름을 나타내는 블록도.

제4도는 디스크 저장 노드의 데이터 흐름을 나타내는 블록도.

제5도는 접속이 이루어지도록 인에이블시키기 위한 제어 메시지의 흐름을 도시.

제6도는 재생이 일어나도록 인에이블시키기 위한 제어 메시지의 흐름을 도시.

제7도는 상기 매체 스트리머와 클라이언트 제어 시스템 사이에 존재하는 인터페이스를 도시.

제8도는 상기 매체 스트리머를 작동시키는데 사용되는 다수의 소프트 키들을 보여주는 디스플레이 패널.

제9도는 제8도상의 적재 소프트 키를 선택하자마자 보여지는 적재 선택 패널을 도시.

제10도는 제8도의 일괄 키가 선택될 때 보여지는 일괄 선택 패널.

제11도는 클라이언트 제어 시스템 및 매체 스트리머 사이에 존재하는 몇몇 클라이언트/서버 관계들을 도시.

제12도는 비디오 데이터를 액세스하고 이 데이터를 하나 이상의 출력 포트들에 공급하는데 사용되는 종래기술을 도시.

제13도는 통신 노드의 캐쉬 메모리에 포함된 단일 비디오 세그먼트에 어떻게 다수의 비디오 포트들이 액세스할 수 있는가를 보이는 블록도.

제14도는 디스크 저장 노드상의 캐쉬 메모리에 포함된 단일 비디오 세그먼트에 어떻게 다수의 비디오 포트들이 직접 액세스할 수 있는가를 보이는 블록도.

제15도는 본 발명에 이용된 메모리 할당 스킴을 도시.

제16도는 비디오 1을 위한 세그먼트화된 논리적 파일을 도시.

제17도는 비디오 1의 여러 세그먼트들이 다수의 디스크 드라이브들에 걸쳐 어떻게 스트라이프화 될 것인가를 도시.

제18도는 저장 노드와 크로스 바 스위치 사이의 종래 스위치 인터페이스를 도시.

제19도는 저장 노드에 대해 연장된 출력 대역폭을 제공하기 위하여 제18도에 보인 종래의 스위치 인터페이스가 어떻게 변경되는가를 도시.

제20도는 비디오 출력 버스로의 일정한 비디오 출력을 보장하기 위한 프로시저를 보이는 블록도.

제21도는 디지털 비디오 데이터를 아날로그 비디오 데이터로 변환하기 위해 사용되는 비디오 어댑터의 블록도.

제22도는 제21도의 비디오 어댑터 카드를 제어하기 위해 SCSI버스 명령들이 사용되도록 하는 제어 모듈들을 보이는 블록도.

* 도면의 주요부분에 대한 부호의 설명

10 : 매체 스트리머 12 : 저 대기시간 스위치

14 : 통신 노드 16 : 저장 노드

18 : 제어 노드 160,162 : 디스크 파일

166 : 파일 시스템 168 : 매체 서버

170 : 비디오 드라이버 180,182 : 디스크

184 : 캐쉬 버퍼 190,192 : 디스크 드라이버

198 : 캐쉬 버퍼 194,196 : 비디오 포트

202,204,206 : FIFO 버퍼들 210 : 버스

212 : 어댑터 214 : 로컬 버퍼

[발명의 상세한 설명]

[관련된 특허출원에 대한 상호참조(cross reference)]

본 출원은 다음의 미국 특허출원들과 관련되어 있습니다.

발명의 명칭이 비디오에 최적화된 매체 스트리머(Video Optimized Media Streamer)이고 발명자는 W.R.벨크냅(Belknap)등인 1994년 9월 8일자 제 08/302,625호 출원.

발명의 명칭이 등시적 데이터 스트림들을 생성하기 위해 비디오에 최적화된 매체 스트리머 (Video Optimized Media Streamer for GeNerating Isochronous Data Streams)이고 발명자는 W.R.벨크냅(Belknap)등인 1994년 9월 8일자 제 08/302,616호 출원.

발명의 명칭이 비디오에 최적화된 매체 스트리머의 데이터 흐름 구조(Video Optimized Media Streamer Data Flow Architecture) 이고 발명자는 M.헨리(Henley)등인 1994년 9월 8일자 제08/302,626호 출원.

발명의 명칭이 캐쉬 관리를 사용하는 비디오에 최적화된 매체 스트리머(Video Optimized Media Streamer with Cache Management) 이고 발명자는 W.R.벨크냅 등인 1994년 9월 8일자 제08/302,619호 출원.

발명의 명칭이 비디오에 최적화된 매체 스트리머 사용자 인터페이스(Video Optimized Media Streamer User Interface)이고 발명자는 A. 색시나(Saxena)등인 1994년 9월 8일자 제 08/303,190호 출원.

[발명의 요약]

본 발명은 멀티미디어 데이터를 전달하는 시스템에 관한 것으로서, 특히 버퍼링을 최소화하면서(minimal buffering) 다수의 단말기들에게 동시에 비디오를 제공 하는 대화형 비디오 서버 시스템(interactive video server system)에 관한 것이다.

[발명의 배경]

영화나 비디오의 재생(playing)은 오늘날 다소 오래된 기술을 사용하여 이루어지고 있다. 주요한 저장 매체는 VHS 리코더/플레이어와 같은 아날로그 테이프이며 이보다 좋은 것으로서 텔레비전 스튜디오나 방송사에 의해 사용되며 고품질에 가격도 비싼 DI VTR들까지 존재한다. 상기 기술들에는 많은 문제점이 있다. 그런 문제점들의 예로는: 테이프들을 적재하는데 요구되는 수작업(manual labor)과 기계적 유닛, 테이프 헤드, 테이프 자체의 마모의 파손 및 비용을 들 수 있다. 방송국을 괴롭히는 심각한 제한 중의 하나는 VTR들이 한번에 한가지 기능씩만 순차적으로 수행할 수 있다는 점이다. 각 테이프 유닛의 가격은 75,000달러에서 150,000달러에 이른다.

TV 방송국들은 특별한 광고방송들을 방송국의 표준 프로그램들에 삽입하여 각 도시를 분리된 시장으로 겨냥함으로써, 짧은 영화들에 불과한 광고방송들을 통해 수입을 늘리기를 원한다. 이것은 고가의 디지털 D1 테이프 시스템이나 테이프 로봇(tape robot)들을 사용하더라도 테이프 기술로는 하기 어려운 작업이다.

최종 사용자(end user)에게 멀티미디어 데이터를 전달하는 종래의 방법들은 1) 방송업계의 방법(broadcast industry method) 및 2) 컴퓨터업계의 방법(computer industry method)의 두가지 카테고리로 분류된다. 방송 방법들(영화, 케이블, 텔레비전 망 및 레코드 산업들을 포함)은 일반적으로 아날로그 또는 디지털로 기록된 테이프의 형태로 저장(storage)을 제공한다. 테이프들을 재생하면 등시적(等時的)데이터 스트림(isochronous data streams)들이 생성되고 이들은 방송 산업 설비를 통해 최종 사용자에게로 이동하게 된다. 컴퓨터 방법들은 일반적으로 디스크들 또는 테이프로 보강된 디스크들(disks augmented with tape) 및 DVI, JPEG, MPEG와 같은 압축 디지털 포맷으로 된 레코드 데이터 형태로 저장을 제공한다. 요청이 있으면 컴퓨터들이 비동시적(non-isochronous) 데이터 스트림들을 최종 사용자에게 전달하며, 최종 사용자측에서 하드웨어 버퍼들 및 특별한 응용코드들이 데이터 스트림들을 평활화시켜(smooth) 연속적인 관람 또는 청취(continuous viewing or listening)를 가능하게 한다.

비디오 테이프 서브시스템들은 그 기억 매체의(낮은) 가격 덕분에 컴퓨터 디스크 서브시스템들에 비해 전통적으로 가격면에서 우위를 보여왔다. 그러나 비디오 테이프 서브시스템들은 테이프 관리, 액세스 대기시간(access latency) 및 상대적으로 신뢰성이 낮다는(iow reliability) 단점을 가진다. 실시간 디지털 압축/압출해제(real-time digital compression/decompression) 기법의 도래와 컴퓨터 저장 장치의 비용 감소에 따라, 비디오 테이프 서브시스템들의 상기 단점들은 그 중요성이 점점 커진다.

비록 컴퓨터 서브시스템들의 비용/성능면에서 복합적인 개선을 보이고 있다 하더라도, 그들은 일반적으로 비디오에 친근한(video friendly) 것으로 여겨지지 않는다. 컴퓨터들은 비동시적(non-isochronous)이라 지칭되는 인터페이스나 프로토콜들에 의해 대개 워크스테이션(workstation) 및 다른 컴퓨터 단말기들에 접촉되고(interface)있다. 멀티미디어 데이터가 최종 사용자에게 평활하게(등시적으로) 전달됨을 보장하기 위해 컴퓨터 시스템들은 그들의 전통적 통신 방법들에 내재한 약점들을 극복하기 위한 특별한 응용 코드 및 대형 버퍼들을 필요로 한다. 또한 등시적 데이터 스트림들을 다루며 고도의 정확성을 가지고 그 데이터 스트림 사이를 스위치하는 멀티미디어 산업의 설비와 양립할 수 있는 인터페이스(compatible interface)들이 컴퓨터들에는 부족하다는 점에서, 컴퓨터들은 비디오에 친근하지 못하다.

비디오 자료(video material)를 디지털 형태로 압축하고, 저장하는데 컴퓨터들의 사용을 도입하면서 텔레비전 방송, 영화 스튜디오 제작, 전화선을 통한 주문형 비디오(video on Demand), 호텔의 유료영화(pay-per-view movies)등과 같은 여러 주요 산업계에서 혁명이 시작되었다. 압축 기술은 100배에서 180배의 압축 비율(compression ratio)로도 허용할만한 결과를 얻을 수 있을 정도로 진보하였다. 이러한 압축 비율은 랜덤 액세스 디스크 기술을 종래의 테이프 시스템들에 대한 매력적인 대안이 될 수 있게 한다.

디지털 디스크 데이터에 대해 랜덤 액세스할 수 있는 능력과 디스크 시스템의 매우 높은 대역폭을 갖는다면, 요구되는 시스템 기능 및 성능은 디스크 기술의 성능 및 하드웨어 비용, 확장성(extendability)의 범위내에 있다. 과거에는 비디오나 영화들을 저장하기 위해 디스크 파일들을 사용하는 것은 그 저장 장치의 가격 때문에 생각조차할 수 없었다. 근래에 와서 그 가격이 상당히 감소하였다.

MPEG 표준들을 사용하여 압축된 비디오 데이터를 이용하는, 새로 출연하는 많은 시장들에 있어서, 가격면에서 효율적인 방식으로(cost effective manner) 비디오 데이터가 저장될 수 있는 여러 방법들이 존재한다. 본 발명은 서로 다른 많은 성능 요구사항에 대해 계층적인 해결책(hierarchical solution)을 제공하여, 시장의 요구사항(market requirement)들을 만족시키도록 개별화(customized)될 수 있는 모듈라 시스템기법(modular system approach)을 형성한다.

[발명의 개요]

본 발명은 멀티미디어 환경에서 그 산업을 위한 전통적 인터페이스들을 통해 등시적 데이터 스트림을 전달할 수 있게 하는 비디오에 친근한 컴퓨터 서브시스템을 제공한다. 본 발명에 따른 매체 스트리머는 등시적 데이터 스트림의 전달을 위해 최적화되며 ATM(비동기 전송 모드)(Asynchronous Transfer Mode) 기법으로 데이터를 새로운 컴퓨터 네트워크에 흘러보낼 수 있다. 본 발명은 시스템 제어용 VTR (비디오 테이프 녹화기)(Video Tape Recorder)과 비슷하게(metaphor) 제공하면서도 비디오 테이프의 단점을 없앤다. 본 발명의 시스템은 다음과 같은 특징들을 제공한다; 최종 사용자에게 독립적으로 제어되는 데이터 스트림들을 1개에서 1000개 까지 전달할 수 있는 스케일 가능성(축소, 확대가능성)(scalability); 단일 데이터 사본으로부터 많은 등시적 데이터 스트림들을 전달

할 수 있는 능력; 혼합된(mixed) 출력 인터페이스들; 혼합된 데이터 속도들; 간단한 개방 시스템(open system) 제어 인터페이스; 자동화 제어 지원(automation control support); 기억 계층 지원(storage hierarchy support); 및 전달되는 스트림당 낮은(전송) 비용.

본 발명에 따른 매체 스트리머는 최소한 하나의 비디오 표현(video presentation)의 디지털 표시(digital representation)를 저장하기 위한 대용량 기억 장치(mass storage)를 구비한, 최소한 하나의 저장 노드(storage node)를 포함한다. 이 대용량 기억장치는 다수의 대용량 기억 장치 유닛(mass storage unit)들로 구성된다. 상기 최소한 하나의 비디오 표현은 그 전체를 나타내는데 시간 T가 소요되며, N개의 데이터 블록들로서 저장된다. 각 데이터 블록은 상기 비디오 표현의 대략 T/N 기간동안에 대응하는 데이터를 저장한다. 상기 매체 스트리머는 또한 다수의 통신 노드(communication node)들을 포함하는데, 각 통신 노드는 비디오 표현의 디지털 표시를 수신하기 위해 최소한 하나의 저장 노드의 출력에 연결된 최소한 하나의 입력 포트를 구비한다. 다수의 통신 노드들 각각은 다수의 출력 포트들을 더 포함하며, 각 출력 포트들은 디지털 표시의 소비자에게 하나의 데이터 스트림으로서 디지털 표시를 전송한다.

본 발명에 의하면 상기 디지털 표시의 N개 데이터 블록들은 X개의 스트라이프(stripe)들로 분할되는데, 데이터 블록들 1, X+1, 2*X+1, ... 등은 X개의 스트라이프들중 첫 번째 스트라이프와 연관되며 데이터 블록들 2, X+2, 2*X+2, ... 등은 X개의 스트라이프들중 두 번째 스트라이프와 연관되고, X개 스트라이프들 각각은 다수의 대용량 기억 장치 유닛들중 서로 다른 유닛에 저장된다.

다수의 대용량 기억 장치 유닛들은 바람직하게는 비디오 표현의 디지털 표시에 대한 단일 사본(single copy)를 저장한다. X개의 스트라이프들은 다수의 데이터 스트림들이 동시에 N개의 데이터 블록들중 동일한 블록을 전달할 수 있게 하는 방식으로 판독되거나, 다수의 데이터 스트림들이 동시에 N개의 데이터 블록들중 서로 다른 데이터 블록을 전달할 수 있게 하는 방식으로 판독된다.

X값은 다음 수식에 따라서 결정된다: $X = \text{maximum}(r*n/d, r*m/d)$ 여기서 r은 데이터 스트림의 명목상 데이터 속도(nominal data rate)이고, n은 상기 명목상 데이터 속도로 동시에 출력되는 데이터 스트림들의 최대 개수이며, d는 대용량 기억 장치 유닛들중 하나의 유효 출력 데이터 속도이며, m은 N개 데이터 유닛들중 최소한 하나를 저장하는 모든 대용량 기억 장치 유닛들로부터 명목상 데이터 속도로 동시에 출력되는 데이터 스트림들의 최대 개수이다.

[본 발명의 상세한 설명]

[용어해설]

이하의 설명에서 아래에 서술된 바와 같은 많은 용어들이 사용된다.

AAL-5 ATM적응층-5 (ATM ADAPTATION LAYER-5): 데이터 전송에 적합한 ATM 서비스의 클래스(class)를 지칭함.

ATM 비동기 전송 모드(ASYNCHRONOUS TRANSFER MODE): 근거리 네트워크나 광역 네트워크(local or wide area network) 또는 둘다에 사용될 수 있는 고속 스위칭 및 전송 기술. 데이터 및 영상/음성(video/audio) 모두를 전달할 수 있게 설계되었다.

Betacam 전문가 품질의 아날로그 비디오 포맷.

CCIR 601 디지털 텔레비전을 위한 표준 해상도. 수평적으로 2:1로 부표본화된 색도(chrominance)와 휘도(luminance)를 갖는 720x840 (NTSC 경우) 또는 720x576(PAL 경우)

CPU 중앙처리장치(CENTRAL PROCESSING UNIT): 컴퓨터 구조에서 컴퓨터 명령어들을 처리하는 주요 엔티티(entity).

CRC 순환 잉여 검사(CYCLIC REDUNDANCY CHECK): 데이터 에러검출 스킴의 하나.

D1 CCIR 601을 따른 디지털 비디오 레코딩 포맷. 19mm 비디오 테이프에 녹화함. D2 SMPTE 244M을 따르는 디

지털 비디오 레코딩 포맷. 19mm비디오 테이프에 녹화함. D3 SMPTE 244M을 따르는 디지털 비디오 레코딩 포맷. 1/2인치 비디오 테이프에 녹화함.

DASD 직접 액세스 저장 장치(DIRECT ACCESS STORAGE DEVICE): 주소 지정이 될 수 있는 임의의 온라인 데이터 저장 장치 또는 CD-ROM 플레이어는 DASD 이다. 자기 디스크 드라이브와 동의어로 쓰인다.

DMA 직접 메모리 액세스(DIRECT MEMORY ACCESS): 컴퓨터 구조에 있어, CPU가 데이터를 옮길 것을 요구하지 않는 데이터 이동 방법.

DVI 상대적으로 낮은 품질의 디지털 비디오 압축 포맷으로서, 보통 CD-ROM 디스크에서 컴퓨터 스크린으로 비디오를 재생하는데 사용된다.

E1 유럽판 T1(European equivalent of T1)

FIFO 선입선출(FIRST IN FIRST OUT): 먼저 온 것을 먼저 처리하는 방식으로 동작하는 대기행렬(Queue) 처리 방법.

GenLock 다른 비디오 신호에 동기화시키는 프로세스(process)를 지칭함. 비디오의 컴퓨터 캡처(computer capture)에서, 비디오 신호의 스캐닝 파라미터(scanning parameter)들과 디지털화 프로세스(digitizing process)를 동기화시키는 것이 요구된다.

I/O 입력/출력

Isochronous 등시적(等時的). 시간에 민감하여 인터럽트 없이(바람직하게는)전송되는 정보를 서술하는데 사용된다. 실시간으로 전송되는 비디오 및 오디오 데이터는 등시적이다.

JPEG 공동 사진 전문가 그룹(JOINT PHOTOGRAPHIC EXPERT GROUP): 국제 표준 기구(International Standards Organization)가 후원하는 운영 위원회로서, 컴퓨터 시스템에서 사용되는 정적 영상(static image)들의 디지털 압축을 위해 제안된 국제 표준을 규정한다.

KB 킬로 바이트(KILO BYTES): 1024 바이트.

LAN 근거리 통신망(LOCAL AREA NETWORK): 단말기들, 컴퓨터들, 주변기기들을 대략 1마일 이하의 거리내에서 연결 꼬임쌍(twisted pair)이나 동축 케이블 또는 광섬유 케이블을 통한 고속 전송.

LRU 최근 최소 사용(LEAST RECENTLY USED).

MPEG 영화 전문가 그룹(MOTION PICTURE EXPERTS GROUP): 국제 표준기구가 후원하는 운영 위원회로서 동 영상/음성(motion video/audio)의 디지털 압축/압축 해제를 위한 표준을 규정한다. MPEG-1은 초기의 표준이며 지금도 사용된다. MPEG-2는 차세대 표준이 될 것이며 디지털이고, 유연성이 있으며, 스케일가능한(scaleable)비디오 전송을 지원할 것이다. MPEG-2는 다중 해상도, 다중 비트 속도, 다중 전달 메카니즘을 지원할 것이다.

MPEG-1, MPEG-2 MPEG참조.

MRU 최근 최대 사용(MOST RECENTLY USED).

NTNU 다음 사용에 가장 가까운 시간(MOST TIME NEXT USE).

NTSC format 국제 텔레비전 표준 위원회(NATIONAL TELEVISION STANDARDS COMMITTEE): 미국 및 일본에서 표준이 되는 칼라 텔레비전 포맷.

PAL format 위상 교대 라인(PHASE ALTERNATION LINE): 프랑스를 제외한 유럽에서 표준이 되는 칼라 텔레비전 포맷.

PC 개인용 컴퓨터(PERSONAL COMPUTER): 가정이나 사업에서 사용할 수 있는 상대적으로 값싼 컴퓨터.

RAID 저가 디스크들의 중복 어레이(REDUNDANT ARRAY OF INEXPENSIVE DISKS): 대역폭 출력을 증가시키고 중복 백업(backup)을 제공하기 위해, 일렬로 동작하는(working in tandem) 여러개의 자기 디스크 또는 광 디스크들을 사용한 저장 장치의 배열.

SCSI 소형 컴퓨터 시스템 인터페이스(SMALL COMPUTER SYSTEM INTERFACE): 주변 장치들 및 그 제어기들을 컴퓨터에 연결하기 위한 산업계의 표준.

SIF 소스 입력 포맷(SOURCE INPUT FORMAT): CCIR 601 해상도의 1/4.

SMPTE 영화 및 텔레비전 엔지니어회(SOCIETY OF MOTION PICTURE TELEVISION ENGINEERS).

SSA 직렬 저장 장치 구조(SERIAL STORAGE ARCHITECTURE): 주변 장치들 및 그 제어기들을 컴퓨터들에 연결하기 위한 표준. SCSI를 대체 할 수 있음.

T1 1.544Mb/초 비트 속도를 가진 전화망에 대한 디지털 인터페이스.

TCP/IP 전송 제어 프로토콜/인터넷 프로토콜(TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL): 미국 국방부에 의해 개발된, 비유사 기종 컴퓨터들(dissimilar computers)을 네트워크를 통해 연결하기 위한 프로토콜 세트.

VHS 수직 나선 스캔(VERTICAL HELICAL SCAN): 자기 테이프에 아날로그 영상을 레코딩하는 일반적인 포맷.

VTR 비디오 테이프 녹화기(VIDEO TAPE RECORDER): 자기 테이프에 영상을 레코딩하는 장치.

VCR 비디오 카세트 녹화기(VIDEO CASSETTE RECORDER): VTR 과 동일.

[A. 일반적 구조(General Architecture)]

비디오에 최적화된 스트림 서버 시스템(10) (이하 매체스트리머라 한다)이 제1도에 보여지며, 이 시스템은 스케일 가능성(scaleability), 높은 가용성(high availability) 및 구성의 융통성(configuration flexibility)을 제공하기 위해서 4개의 구조적으로 분리된 구성요소(architecturally distinct component)들을 포함한다. 주요 구성요소는 다음과 같다:

- 1) 저 대기시간 스위치(low latency switch)(12): 통신 노드들(14), 하나이상의 저장 노드들(16, 17) 및 하나 이상의 제어 노드들(18) 사이에 제어 노드들(18) 사이에 데이터와 제어 정보를 전달하는 것이 주요 태스크인 하드웨어/마이크로코드로된 구성요소.
- 2) 통신 노드(14): 외부적으로 정의된, 통상 방송 업계에 친숙한 인터페이스(NTSC, PAL, D1, D2 등)를 통해 재생(playing) (데이터를 등시적으로 전달) 또는 레코딩(recording) (데이터를 등시적으로 수신)을 가능케 하는 것이 주요 태스크인 하드웨어/마이크로코드로된 구성요소. 각 통신 노드의 출력에 연결된 다수의 비디오 포트(15)에 포함되어 있는 비디오 카드안에 디지털 대 비디오(digital-to-video) 인터페이스가 구현되어 있다.
- 3) 저장 장치 노드(이하 저장 노드라 하자)(16,17): 디스크와 같은 저장 매체 및 그와 관련된 저장 장치 가용성 옵션(storage availability option)들을 관리하는 것이 주요 태스크인 하드웨어/마이크로코드로된 구성요소.
- 4) 제어 노드(18): 컴퓨터 업계에 친숙한 외부적으로 정의된 서브시스템 인터페이스로부터 제어 명령들을 수신하고 실행하는 것이 주요 태스크인 하드웨어/마이크로코드로된 구성요소.

64개 노드들로 구현하는 전형적인 매체 스트리머는 저 대기시간 스위치(12)로 상호 연결된 31개의 통신 노드들, 31개의 저장 노드들, 2개의 제어 노드들을 포함한다. 이보다 더 작은 시스템이라면 스위치가 없고 통신, 저장 및 제어기능을 지원하는 단일의 하드웨어 노드를 포함할 수 있다. 매체 스트리머(10)의 설계는 고객이 시스템 설치시 소형 시스템에서 대형 시스템으로 확장(grow)할 수 있게 되어 있다. 모든 구성에 있어서, 매체 스트리머(10)의 기능적 능력은 전달되는 스트림의 수와 저장된 멀티미디어 (소요)시간들을 제외하고는 동일하다.

제 1a도에는, 저 대기시간 스위치(12)의 세부가 도시되어 있다. 다수의 회선 스위치 칩 (도시되지 않음)들이 크로스바 스위치 카드들(20)에 연결되어 있으며 이 크로스바 스위치 카드들은 하나의 평면 보드(planar board)(개략적으로 도시됨)를 통해 상호 연결되어 있다. 상기 평면 보드와 카드(20) 하나가 16개 노드 포트들을 가진 저 대기시간 크로스바 스위치를 구성한다. 추가적인 노드 포트들을 구성하기 위해서, 원한다면 높은 가용성을 위한 활동 중복 노드 포트(active redundant node ports)들을 구성하기 위해서 추가적인 카드들(20)이 부가될 수 있다. 저 대기시간 스위치(12)의 각 포트들은 예를 들어 1초에 2메가바이트 속도의 전 이중 통신 채널(full duplex communication channel)을 인에이블시킬 수 있다.

정보는 상기 스위치(12)를 통해 패킷 형태로(in packets)전달된다. 각 패킷은 각 스위치 칩들에서 개별 크로스바 스위치 점들의 스위칭 상태를 제어하는 헤더 부분(header portion) 포함한다. 제어 노드(18)는 다른 노드들(저장 노드(16,17) 및 통신 노드(14))에게 상기 저 대기시간 스위치(12)를 통한 피어-투-피어 동작(peer-to-peer operation)을 인에이블 시키는데 필요한 정보를 제공한다.

제 1b도에서는 테이프 저장 노드(17)의 세부가 도시되어 있다. 이하에서 이해될 수 있는 바와 같이, 테이프 저장 장치 노드(17)는 비디오 표현에 대한 디지털 표시의 저장을 위해 고용량의 저장 기능을 제공한다.

이하에서 이용된 것처럼, 비디오 표현은 디스플레이하기에 그리고/또는 처리하기에 적당한 하나 이상의 영상(image)들을 포함할 수 있다. 비디오 표현은 음성부분(audio portion)을 포함할 수도 있다. 하나 이상의 영상들은 필름, 영화, 또는 만화의 연속된 장면(sequence)들의 순차적 프레임들과 같이 논리적으로 서로 관련 되어 있다. 상기 영상들은 처음에 카메라나 디지털 컴퓨터에 의해 생성되거나 또는 이 두가지의 조합에 의해 생성되었을 수 있다. 음성부분은 연속된 영상들의 디스플레이에 동기화될 수 있다. 이하에 이용된 바와 같이 비디오 표현의 데이터 표시는, 하나 이상의 영상들과 (대개는) 음성을 표시하는데 적합한 임의의 디지털 데이터 포맷일 수 있다. 상기 디지털 데이터는 부호화 및/또는 압축될 수 있다.

다시 제 1b도를 참조하면, 테이프 저장 노드(17)는 테이프 라이브러리 제어기 인터페이스(tape library controller interface)(24)를 포함하며 이 인터페이스는 테이프 라이브러리(26)에 포함된 다수의 테이프 레코드들에 대한 액세스를 가능하게 한다. 또 인터페이스(28)는 SCSI 버스 연결(bus interconnection)을 통해 다른 테이프 라이브러리 들로의 액세스를 가능하게 한다. 내부 시스템 메모리(internal system memory)(30)는 인터페이스들(24,28)중 하나로 부터 수신된 비디오 데이터 또는 DMA 데이터 전송경로(32)를 통한 비디오 데이터의 버퍼링(buffering)을 가능하게 한다. 시스템 메모리 블록(30)은, 테이프 라이브러리 및 파일 관리 동작을 위한 소프트웨어(36)를 포함한 PC(34)의 일부분일 수 있다. 스위치 인터페이스 및 버퍼 모듈(38)(디스크 저장 노드들(16), 통신 노드들(4) 및 제어 노드들(18)에서도 사용됨)은 테이프 저장 노드와 저 대기시간 스위치(12)간의 상호 연결을 가능하게 한다. 즉, 스위치(12)가 패킷의 경로지정(route)하는 데 사용할 수 있도록 전송 데이터를 패킷들로 분할하고 각 패킷에 헤더 부분을 부가하는 책임을 상기 모듈(38)이 맡고 있다. 스위치(12)로부터 패킷을 수신하면, 모듈(38)은 수신된 데이터를 국소적으로 버퍼링하거나(locally buffering) 수신된 데이터에 기타 다른 조작을 하기전에 상기 헤더 정보를 떼어낼 책임이 있다.

테이프 라이브러리(26)로부터의 비디오 데이터는 첫 버퍼링 동작으로 인해 시스템 메모리(30)에 들어간다. 그런 다음 제어 노드(18)로부터의 초기 지시(initial direction)에 응답하여, 상기 비디오 데이터는 저 대기시간 스위치(12)를 통해 디스크 저장 노드(16)로 경로 지정되어(routed) 필요할 때 사실상 즉각적으로 액세스할 준비가 되게 한다.

제 1c도에서는 디스크 저장 노드(16)의 내부가 상세히 보여진다. 각 디스크 저장 노드(16)마다 데이터가 RAID 버퍼 비디오 캐쉬 및 저장 장치 인터페이스 모듈(42)로/부터 데이터가 전송되게 하는, 스위치 인터페이스 및 버퍼 모듈(40)을 포함한다.

인터페이스(42)는 수신된 비디오 데이터를 다수의 디스크들(45)로 전달하되 디스크들상에 데이터들을 준 RAID 방식(quasi-RAID fashion)으로 분포시킨다. RAID 메모리 저장 장치에 대한 세부사항은 종래기술에 알려져 있으며,

1988년 6월 1 내지 3일에 일리노이주 시카고 ACM SIGMOD 컨퍼런스에 있었던 패터슨(patterson)등의 A case for redundant Arrays of Inexpensive Disks(RAID) 109-116페이지에 서술되어 있다.

디스크 저장 노드(16)는 또한 소프트웨어 모듈들(46,48)을 포함하는 내부 PC(44)를 구비하되 여기서 소프트웨어 모듈들(46,48)은 각각 저장 노드 제어 및 비디오 파일과, 디스크 제어 및 디스크(45)상에 저장된 데이터들을 위한 RAID 매핑(mapping)을 제공한다. 사실상, 각 디스크 저장 노드(16)는 테이프 저장 노드(17)보다 더 즉각적인 수준(immediate level)의 비디오 데이터 가용성을 제공한다. 각 디스크 저장 노드(16)는 또한 더욱 신속하게 비디오 데이터를 제공하기 위해서 비디오 데이터에 대한 요청을 수신하자마자 스위치 인터페이스 및 버퍼 모듈(40)의 반도체 메모리에 비디오 데이터를 버퍼링(캐쉬 방식으로)할 수 있다.

일반적으로 저장 노드는 대용량 저장 장치 유닛(또는 대용량 저장 장치 유닛으로의 인터페이스)과 대용량 저장장치 유닛으로부터 판독하거나 그곳에 기록될 데이터를 국소적으로 버퍼링할 수 있는 능력을 포함한다. 저장 노드는 하나 또는 그 이상의 테이프 드라이브들 및/또는 디스크 드라이브들의 형태로 된 순차 액세스 대용량 저장 장치를 포함할 수 있고, 랜덤 액세스 방식으로 액세스되는 하나 또는 그 이상의 디스크 드라이브들 및/또는 반도체 메모리와 같은 랜덤 액세스 저장 장치를 포함할 수도 있다.

제1d도에서는 통신 노드(14) 내부의 구성요소들에 대한 블록도가 보여진다. 앞서의 각 노드들과 유사하게, 통신 노드(14)는 전송할 것과 같이 저 대기시간 스위치(12)와 통신할 수 있게 하는 스위치 인터페이스 및 버퍼 모듈(50)을 포함한다.

비디오 데이터는 스위치 인터페이스 및 버퍼 모듈(50)과 사용자의 단말기(도시되지 않음)에 전송하기 위한 스트림 버퍼 및 통신 인터페이스(52) 사이에서 직접 전송된다. PC(54)는 소프트웨어 모듈들(56,58)을 포함하며 이들은 각각 통신 노드 제어(예를 들면 스트림의 개시/정지 동작들) 및 데이터의 등시적 스트림의 계속적 생성을 제공한다. 스트림 버퍼 및 통신 인터페이스(52)에 대한 부가적 입력(60)은 출력 데이터의 프레임 동기화를 가능하게 한다. 상기 데이터는 자동화 제어 장치(automation control equipment)로부터 수신되며, 이 장치는 스트림 서버(10)의 전체적인 동작을 제어하는 시스템 제어기(64)에 의해 제어된다(제1도 참조). 시스템 제어기(64)는 사용자 제어 형 세트 탑 박스들(user control set top boxes)(65)로부터의 입력에 응답하여, 매체 스트리머(10)가 요청된 비디오 표현에 액세스할 수 있도록 명령들이 생성되게 한다. 시스템 제어기(64)에는 또한 사용자 인터페이스 및 디스플레이 장치(66)가 제공되어, 사용자로 하여금 하드 또는 소프트 버튼(hard or soft button)들과 같은 명령들 및 비디오 표현들의 식별(identification), 비디오 표현들의 스케줄링, 비디오 표현을 재생하는 동안의 제어를 가능하게 하는 다른 데이터들을 입력할 수 있게 한다.

각 제어 노드(18)는 PC로서 구성되며(configured) 저 대기시간 스위치(12)와 인터페이스하기 위한 스위치 인터페이스 모듈을 포함한다. 각 제어 노드(18)는 시스템 제어기(64)로부터의 입력들에 응답하여, 원하는 상호연결이 저 대기시간 스위치(12)를 통해 생성될 수 있도록 통신 노드들(14) 및 저장 장치 노드들(16,17)에게 정보를 제공한다. 또한, 제어 노드(18)는 요청된 비디오 데이터를 하나 또는 그 이상의 디스크 저장 장치 노드들(16)로부터 스테이징(staging)하고, 그 비디오 데이터를 스트림 전달 인터페이스를 통해 사용자 디스플레이 단말기까지 전달하는 소프트웨어를 포함한다. 제어 노드(18)는 또한 저 대기시간 스위치(12)를 통해 보내지는 명령들을 통해서 테이프 및 디스크 저장 노드(16,17) 모두의 동작을 제어한다.

매체 스트리머는 제1도에 보인 바와 같이 3개로 구성된 외부 인터페이스(external interface)들을 가진다. 이 외부 인터페이스들은 다음과 같다:

- 1) 제어 인터페이스: TCP/IP 프로토콜을 실행하는 개방형 시스템 인터페이스(이써넷 LAN, 토큰링 LAN, 직렬 포트, 모뎀 등).
- 2) 스트림 전달 인터페이스: 데이터 스트림들의 전달을 위해 설계된 여러 산업계 표준 인터페이스들 중 하나(NTSC, D1 등).
- 3) 자동화 제어 인터페이스: 스트림 출력들의 정확한 동기화를 위한 산업계 표준 제어 인터페이스들의 모임(GenLock, BlackBurst, SMPTE Clock 등).

응용명령(Application command)들은 상기 제어 인터페이스를 거쳐 매체 스트리머(10)에 발행된다(issued).

데이터 적재 명령이 발행되면, 제어 노드는 들어오는 데이터 파일들을 세그먼트들로 (즉 데이터 블록들로) 쪼개고 이들을 하나 또는 그 이상의 저장 노드들에 걸쳐 분포시킨다. 밀도 및 데이터의 동시 사용자 수가 저장 노드들(16,17)에 대한 데이터 배치에 영향을 미친다. 밀도 및/또는 동시 사용자 수의 증가는 용량 및 대역폭에 비해 더 많은 저장 노드들을 사용하는 것을 암시한다.

최종 사용자에게로의 데이터 스트리밍을 시작하기 위해서 제어 인터페이스를 통해 명령들이 발행되면 제어 노드(18)는 적절한 통신 노드(14)를 선택하고 활성화시키며, 이 통신 노드에게 저장 노드들(16,17)상에 데이터 파일 세그먼트들의 위치를 표시하는 제어 정보를 전달한다. 통신 노드(14)는 관련될 필요가 있는 저장 노드들(16,17)을 활성화시키고, 저 대기시간 스위치(12)를 거쳐 보내지는 명령 패킷들을 통해 이들 노드들과의 통신을 개시함으로써 데이터 이동을 시작한다.

데이터는 저대기시간 스위치(12) 및 저스트 인 타임(just in time) 스케줄링 알고리즘을 통해 디스크 저장 노드(16)들 및 통신 노드(14)들 사이에서 이동된다. 스케줄링 및 데이터 흐름 제어에 사용되는 기술은 이하에서 더욱 상세히 설명된다. 통신 노드 인터페이스(14)로부터 방출되는 데이터 스트림은 디스크 저장 노드들(16)로부터 다중화되어(multiplexed), 단일 통신 노드 스트림이 각 디스크 저장 장치 노드(16)의 용량 및 대역폭 일부를 사용할 수 있게 된다. 이런 방식으로 많은 통신 노드들(14)이 디스크 저장 노드(16)들상의 동일한 또는 다른 데이터에 대한 액세스를 다중화할 수 있다. 예를 들어 매체 스트리머(10)는 통신 노드들(14)의 풀(pool)로부터 1500개의 개별적으로 제어되는 최종 사용자 스트림들을 제공할 수 있는데, 이 스트림들 각각은 디스크 저장 노드들(16)에 걸쳐 퍼져 있는 단일 멀티미디어 파일에 대한 액세스들을 다중화한다. 이 능력을 단일 사본 다중 스트림(single copy multiple stream)이라 한다.

제어 인터페이스를 통해 수신된 명령들은 두가지 구분되는 카테고리들로 실행된다. 데이터를 관리할 뿐 스트림 제어에는 직접적으로 관련되지 않은 명령들은 낮은 우선순위(low priority)로 실행된다. 이는 응용으로 하여금 최종 사용자에 대한 데이터 스트림들 전달을 방해하지 않고도 매체 스트리머(10)에 새로운 데이터를 적재할 수 있게 한다.

스트림 전달(즉 출력)에 영향을 미치는 명령들은 높은 우선순위(high priority)로 실행된다.

제어 인터페이스 명령들이 제2도에 보여진다. 매체스트리머(10)내에서 데이터를 적재하고 관리하기 위한 낮은 우선순위 데이터 관리 명령들로는 VS-CREATE, VS-OPEN, VS-READ, VS-WRITE, VS-GET_POSITION, VS-SET_POSITION, VS-CLOSE, VS-RENAME, VS-DELETE, VS-GET_ATTRIBUTES 및 VS-GET_NAMES 이 포함된다.

스트림 출력들을 개시하고 관리하기 위한 높은 우선순위 스트림 제어 명령들로는 VS-CONNECT, VS-PLAY, VS-RECORD, VS-SEEK, VS-PAUSE, VS-STOP 및 VS-DISCONNECT가 포함된다. 제어 노드(18)는 요청들이 실행될 수 있음을 확실히 하기 위해 스트림 제어 명령들을 모니터한다. 매체 스트리머(10)의 가능 출력(capability)이 초과된 경우, 제어 노드(18)의 이런 입장 제어(admission control) 기능(facility)은 스트림들을 개시하는 요청을 거절할 수 있다. 이는 여러 상황들에서 발생할 수 있다:

- 1) 시스템내 몇몇 구성요소가 고장이 나서 최대 동작(maximal operation)을 수행할 수 없는 경우;
- 2) 하나의 데이터 파일에 대해 지정된 동시 스트림들의 개수(VS-CREATE 명령의 파라미터들에 의해 지정된 바와 같이)가 초과된 경우; 및 3) 인스톨레이션 구성(installation configuration)에 의해 지정된 바와 같이, 시스템으로부터 나오는 동시 스트림들의 지정된 개수가 초과된 경우.

통신 노드들(14)은 각각은 잠재적으로 서로 다른 대역폭(스트림) 능력 및 물리적 정의를 갖는, 하나의 이종 그룹(heterogeneous group)으로 다루어진다. VS-CONNECT 명령은 매체 스트리머(10)로 하여금 통신 노드(14)를 할당하게 하고 동시적 데이터 스트림 전달을 가능하게 하는 관련 대역폭의 일부 또는 전부를 할당하도록 한다. 예를 들어 매체 스트리머(10)는 압축되지 않은 데이터 스트림(들)을 통신 노드(들)(14)을 통해 270M비트/초의 속도로 재생할 수 있으며, 동시에 다른 통신 노드들(14)에서는 훨씬 낮은 데이터 속도로(대개 1 내지 16M비트/초) 압축된 데이터 스트림(들)을 재생할 수 있다.

저장 노드들(16,17)은 각각 잠재적으로 서로 다른 대역폭(스트림) 능력 및 물리적 정의를 갖는 하나의 이종 그룹들로서 관리된다. VS-CREATE 명령은 매체 스트리머(10)로 하여금 멀티미디어 파일 및 그에 관련된 메타데이터(metadata)를 위해 하나 또는 그 이상의 저장 장치 노드들(16,17)내에 있는 저장 장치를 할당하게 한다. VS-

CREATE 명령은 스트림 밀도 및 요구되는 최대 동시 사용자수 모두를 지정한다.

3개의 부가적인 명령들(VS-CONNECT_LIST, VS-PLAY-AT-SIGNAL 및 VS-RECORD-AT-SIGNAL)이 방송 업계의 자동화 제어 시스템들(automation control system)을 지원한다. VS-CONNECT_LIST 명령은 응용들로 하여금 서브시스템에 대한 단일 명령으로 일련의 재생 명령들(a sequence of play commands)을 지정할 수 있도록 한다. 매체 스트리머(10)는 각 재생 명령을 마치 제어 인터페이스에서 발행된 것처럼 실행할 뿐 아니라 하나의 스트림 전달과 그 다음 스트림 전달 사이에서 이음새없이(seamlessly)전이시킬 것이다. 본보기적인 순차는 다음과 같다:

1) 제어 노드(18)는 파일 1, 파일 2 및 파일 3의 전부 또는 일부가 순차적으로 재생되어야 함을 지시하는 재생 부명령(play subcommand)들로 된 VS-CONNECT_LIST 명령을 수신한다. 제어 노드(18)는 파일들의 최대 데이터 속도를 결정하고 그 자원을 통신 노드(14)에 할당한다. 할당된 통신 노드(14)는 상세한 재생 목록을 수신하고 등시적 스트림의 전달을 개시한다.

2) 파일 1의 전달이 종료될 무렵, 통신 노드(14)는 파일 2의 전달을 개시하지만 이들 상기 노드의 출력 포트로 보낼 수는 없다. 파일 1이 종료되거나 자동화 제어 인터페이스로부터 신호가 발생하면, 통신 노드(14)는 출력 포트를 제1스트림으로부터 제2스트림으로 스위치한다. 이는 1/30초 또는 하나의 표준 비디오 프레임 시간내에 행해진다.

3) 통신 노드(14)는 파일 1에 관련된 자원들을 할당 해제(deallocate)한다.

VS-PLAY-AT-SIGNAL 및 VS-RECORD-AT-SIGNAL은 외부의 자동화 제어 인터페이스로부터의 신호들로 하여금 재생 및 기록 동작들을 위한 데이터 전송을 비디오 프레임 경계로 정확하게 할 수 있게 한다. 앞서의 예에서 VS-CONNECT_LIST 명령은, 외부의 자동화 제어 인터페이스 신호에 기초하여 파일 1에서 파일 2로 전이케 하기 위한 PLAY-AT-SIGNAL 부명령을 포함한다. 만약 상기 부명령이 PLAY-AT-SIGNAL 대신 VS-PLAY 라면, 상기 전이는 단지 파일 1의 전송이 종료된 경우에만 일어날 것이다.

매체 스트리머(10)가 실행하는 다른 명령중에는 기억 계층(storage hierarchies)을 다루는 능력을 제공하는 것이 있다. 그런 명령으로서 VS-DUMP, VS-RESTORE, VS-SEND, VS-RECEIVE 및 VS-RECEIVE_AND_PLAY가 있다. 이들 각각은 두 개의 외부적으로 정의된 계층적 엔터티들과 저장 노드들(16)간에 하나 이상의 멀티미디어 파일들이 이동될 수 있게 한다.

1) VS-DUMP 및 VS-RESTORE 명령은 제어 노드(18)가 액세스할 수 있는 디스크 저장 장치 노드들(16) 및 테이프 저장 장치 유닛(17)들 간의 데이터 이동을 가능하게 한다. 데이터 이동은 제어용 응용(controlling application)에 의해 개시되거나 제어 노드(18)에 의해 자동적으로 개시될 수 있다.

2) VS-SEND 및 VS-RECEIVE는 멀티미디어 파일을 다른 매체 스트리머로 전송하는 방법을 제공한다. 임의적으로(optionally), (멀티미디어 화일을) 수신하는 매체 스트리머는 파일 전체를 기다리지 않고도 사전에 할당된 통신 노드에 입력되는 파일을 즉시 재생 할 수 있다.

본 매체 스트리머 구조에서 규정된 모듈라 설계 및 기능 세트(modular design and function set)에 부가하여, 상당히 낮아진 가격으로 등시적 데이터 전송을 하기 위해 데이터 흐름도 최적화된다.

특히: 1) 저 대기시간 스위치의 대역폭은 부착된 노드들의 대역폭을 초과한다; 노드들간의 통신은 거의 블록킹되지 않는다(non-blocking);

2) 프로세서 메모리로의 데이터 이동은 회피되며 더 넓은 대역폭이 제공된다;

3) 데이터 처리는 회피된다; 고가의 처리 장치들은 제거된다;

4) 커다란 데이터 캐쉬를 쓰지 않도록 데이터의 이동은 주의깊게 스케줄된다.

전통적인 컴퓨터 용어들에 있어서, 매체 스트리머(10)는 상호 연결된 어댑터(interconnected adapter)들의 시스템으로서 동작 하며, 이 어댑터들은 저 대기시간 스위치(12)를 통해 피어-피어 데이터 이동(peer-peer data movement)을 수행하는 능력을 가진다. 저 대기시간 스위치(12)는 호스트 컴퓨터(host computer)의 개입없이도 데이터 저장 장치에 대한 액세스를 할 수 있고 한 어댑터의 메모리로부터 다른 어댑터의 메모리로 데이터 세그먼트들을 이동시킬 수 있다.

[B. 등시적 전달을 위한 압축 디지털 비디오 데이터의 계층적 관리(Hierarchical management of digital compressed video data for isochronous delivery)]

매체 스트리머(10)는 계층적 저장 요소(hierarchical storage element)들을 제공한다. 매체 스트리머는 매우 작은 비디오 시스템에서 매우 큰 시스템으로의 확대가능성(scaleability)이 허용되는 설계로 되어 있다. 이는 또한 저장 장치 관리에 유용성을 제공하여 주문형 비디오(video on Demand), 준 주문형 비디오(Near Video on Demand) 광고방송 삽입(commercial insertion), 고품질 비압축 비디오의 저장, 캡처 및 재생(storage, capture and playback) 기능을 만족시키는데 필요한 여러 요구사항들을 따라 저장 장치 관리가 개조될 수 있게 한다.

[B1. 테이프 저장 장치(tape storage)]

매체 스트리머(10)에서, 비디오 표현들은 최종 사용자에 의해 요구되는 매우 낮은 데이터 속도로 재생된후 고성능 디지털 테이프에서 디스크로 이동된다. 이런 방식으로 비디오 시간중 최소량만이 디스크 서브시스템에 저장된다. 만약 시스템이 준 주문형 비디오(Near video on Demand) 라면, 각 영화는 예를 들어 한 번에 5분정도만 디스크 저장 장치내에 있으면 된다. 이는 전형적인 2시간 짜리 영화마다 5분간의 세그먼트 22개만을 요구한다. 그 결과 하나의 비디오 표현을 위한 총 디스크 저장 장치 요구라 감소하는데, 이는 어느 때라도 디스크상에 그 비디오 표현 전체가 유지되지는 않기 때문이다. 상기 비디오 표현중 단지 재생되는 부분만이 디스크 파일상에 존재할 필요가 있다.

다시 말해서 하나의 비디오 표현이 그 전체가 존재하는데 시간 T가 필요하며 N개 데이터 블록들을 가진 디지털 표시로서 저장된다면, 각 데이터 블록은 상기 비디오 표현의 대략 T/N시간에 대응하는 비디오 표현 부분을 저장한다. N개 데이터 블록들중 가장 마지막 데이터 블록은 T/N 시간보다 덜 저장할 수도 있다.

시스템에 대한 수요가 늘어나고 스트림들의 수가 증가됨에 따라 통계적인 평균으로는 스트림들 요청중 같은 영화에 대한 것이 약 25% 이지만, 다른 어떤 서브-세컨드(sub-second)시간 구간에서는 그 서브-세컨드 동안의 수요들중 50% 이상이 15개 영화 세그먼트들의 그룹내에 속하도록 시청자들이 분포할 것이다.

상기 수요를 만족시킬 수 있는 가장 적당한 기술을 이용하는 것이 본 발명의 일면이다. 랜덤 액세스 카트리지 적재기(random access cartridge loader)(IBM 사에서 생산한 것과 같은)는 테이프당 기억 용량이 크며 드로어(drawer)당 100개 테이프들을 기계적으로 로봇을 이용하여 적재하고 드로어당 최대 2개 테이프 드라이브들을 포함하는 디지털 테이프 시스템이다. 그 결과 주문형 영화 시스템(movie-on-demand system)을 위한 효율적인 테이프 라이브러리가 될 수 있다.

그러나 본 발명은 또한 매우 저렴한 디지털 테이프 저장 장치 라이브러리 시스템으로 하여금 영화들을 대량으로 저장할 수 있게 하고, 또한 수요가 적은 영화들은 테이프에서 직접 스피드 정합 버퍼들(speed-matching buffers)로 재생된 후 비디오 압축 해제 및 분배 채널(video decompression and distribution channel)상에 있게 한다. 임의의 비디오 시스템에 대해 계층적 테이프 저장 장치를 결합함이 갖는 두 번째 장점은, 디스크 하나가 작동하지 않게 될 때 그 디스크 상에 저장된 어떠한 영화에 대해서도 신속하게 백업(backup)을 제공한다는 점이다. 통상적인 시스템은 하나의 디스크 유닛이 고장나면 테이프로부터 영화들이 재적재(reload) 될 수 있는 여분의(spare) 디스크를 유지할 것이다. 이는 통상 RAID 또는 그와 유사한 (RAID-like) 시스템과 결합될 것이다.

[B2. 디스크 저장 장치 시스템(Disk Storage Systems)]

비디오 스트림들에 대한 수요가 더 높은 단계로 증가함에 따라, 하나의 영화 전체를 디스크에 저장함으로써 비디오 데이터를 계속 테이프에서 디스크로 옮기는데 요구되는 시스템 성능 오버헤드를 절약한다면 더욱 효율적이 된다. 통상적인 시스템은 여전히 테이프에 저장되는 영화들의 라이브러리를 포함하는데, 이는 라이브러리에 있는 영화들의 통상적인 숫자가 특정시간에 재생될 영화들의 수보다 10배에서 100배 정도 크기 때문이다. 한 명의 사용자가 특정한 영화를 요청하면, 그 영화의 세그먼트들이 디스크 저장 노드(16)에 적재되고 그곳에서 부터(재생이) 시작된다.

동일한 영화를 보고싶어하는 사용자가 많이 있는 경우라면 그 영화를 디스크상에 계속 유지하는 것이 유리하다. 이런 영화들은 통상 그 주의 인기 영화(hot" movies)들이며 피크 시청 시간대(peak viewing hours) 이전에 테이프

로부터 디스크로 사전적재(pre-loaded) 된다. 이는 피크 시간대동안의 시스템 작업부담(workload)을 감소시켜주는데 이바지한다.

[B3. 캐쉬로부터 나오는 영화들(Movies out of cache)]

인기 영화들에 대한 수요가 증가함에 따라서, 매체 스트리머(10)는 MRU에 기초한 알고리즘(MRU-based algorithm)을 통해 중요한 영화들을 캐쉬로 옮길 것을 결정한다. 이는 상당한 양의 캐쉬 메모리를 요구하지만, 비용 대 활동 스트림들 개수의 비율(ratio of cost to the number of active streams) 면에서 볼 때 캐쉬로부터 많은 양이 지원된다는 사실이 스트리머(10)의 전체 비용을 감소시킨다. 비디오 데이터의 성질(nature) 때문에, 또한 어떤 비디오들이 재생중이며 다음은 어떤 데이터가 요구될 것인지 얼마동안 요구될 것인지를 시스템이 항상 미리 알고 있다는 사실 때문에, 캐쉬, 내부 버퍼, 디스크 저장 장치, 테이프 적재기, 버스 성능 등의 이용을 최적화하는 방법들이 사용된다.

저장 매체 전부에 걸쳐 그 내용의 일괄 및 분배를 제어하는 알고리즘들은 요구되는 대역폭들의 넓은 스펙트럼으로 등시적 데이터 전달을 할 수 있게 한다. 등시적 데이터의 전달은 사실상 100% 예측가능한 것이므로, 상기 알고리즘들은, 컴퓨터 업계의 다른 분야들에 사용되는 전통적인 알고리즘들 즉 사용자에게 의해 액세스되는 데이터의 캐쉬가(caching of user-accessed data) 항상 예측되는 것이 아닌 알고리즘들과는 상당히 다르다.

[C. 매체 스트리머의 데이터 흐름 구조(Media streamer data flow Architecture)]

전술한 바와 같이, 매체 스트리머(10)는 TV 세트들 및 LAN, ATM 등과 같은 네트워크를 통해 부착된 세트 탑 박스(set top box)들과 같은 다양한 출력(장치)들에 비디오 스트림들을 전달한다. 저장 장치 용량 및 동시 스트림들의 개수에 대해 요구되는 조건들을 만족 시키기 위해서는 다수의 저장 노드 및 통신 노드들로 구성된 분산형 구조(distributed architecture)가 바람직하다. 데이터는 저장 노드들(16,17)에 저장되며 통신 노드들에 의해 전달된다. 통신 노드(14)는 적절한 저장 노드들(16,17)로부터 데이터를 얻는다. 제어 노드(18)는 하나의 시스템 이미지(single system image)를 외부세계에 제공한다. 상기 노드들은 저 대기시간 스위치(12)의 크로스-커넥트(cross-connect)에 의해 연결된다.

각 스트림마다 데이터 속도 및 전달되어야 할 데이터를 예측할 수 있다. 본 발명은 이 예측 가능성을 이용하여, 자원들을 완전히 사용하게 하며 필요한 경우 모든 단계에서 각 스트림을 위한 데이터가 이용될 수 있음을 보증하는 데이터 흐름 구조(data flow architecture)를 구성한다. 저장 노드들(16,17) 및 통신 노드들(14)간의 데이터 흐름을 여러 가지 다른 방식으로 셋업(set up)될 수 있다.

통신 노드(14)는 일반적으로 다중 스트림들의 전달을 책임진다. 통신 노드는 이들 각 스트림들을 위한 데이터를 기다리는(outstanding) 요청들을 가질 수 있으며, 이 요구되는 데이터는 서로 다른 저장 노드들(16,17)로부터 나올 수 있다. 만약 서로 다른 저장 노드들이 동일한 통신 노드에 동시에 데이터를 보내려고 시도한다면, 단지 하나의 저장 노드만이 데이터를 보낼 수 있고 다른 저장 노드는 데이터를 전송할 수 없다(blocked). 이 전송 봉쇄(blockage)는 (봉쇄된)저장 노드들로 하여금 데이터 전송을 재시도하게 야기하므로 스위치 이용도를 저하시킬 뿐 아니라 저장 노드에서 통신 노드로 데이터를 보내는데 필요한 시간이 큰 분산을(large variance) 갖게 한다. 본 발명에서는 서로 다른 저장 노드들(16,17)간에 통신 노드(14)의 입력 포트에 대한 경쟁(contention)이 존재하지 않는다.

요구되는 버퍼링의 양은 다음과 같이 결정될 수 있다: 통신 노드(14)가 요청(request)을 저장 노드(16,17)에 보내고 데이터를 수신하는 데까지 필요한 평균 시간을 결정한다. 이 시간은 저장 노드에 요청을 보내는 시간과 그 응답을 수신하는 시간을 저장 노드가 요청을 처리하는데 필요한 시간에 더함으로써 결정된다. 그 다음은 저장 노드가 디스크로부터 데이터를 판독하는데 필요한 평균 시간과 그 요청의 처리에 관련된 임의의 지연들을 더함으로써 그 요청처리에 요구되는 평균 시간을 결정한다. 이것이 그 요청 처리에 있어서의 대기시간(latency)이다. 요구되는 버퍼링의 양은 스트림 데이터 속도로 상기 대기시간을 보상(cover)하는데 필요한 메모리 저장 장소만큼이다. 후술하는 해결 방법은 대기시간을 줄이고 그럼으로써 요구되는 자원들을 줄이기 위해 매체 스트리머 환경에서 특별한 조건들을 사용한다. 데이터의 모든 단계(예를 들면 저장 노드들 및 통신노드들 안에서)마다, 그전 단계로부터 데이터에 대한 요청을 예측함과 함께 저스트-인-타임 스케줄링(just-in-time scheduling)을 사용함으로써 상기 대기시간은 감소된다.

통신 노드(14)의 입력 포트에 대한 저장 노드들(16,17)의 경쟁은 다음 두가지 기준(criterion)을 사용함으로써 제거

될 수 있다:

1) 저장 노드(16,17)는 특정한 요청을 받자마자 통신 노드(14)에 데이터를 단지 보낸다.

2) 주어진 통신 노드(14)는 저장 노드로부터의 데이터 판독 요청 모두를 직렬화하여(serialize), 통신 노드(14)가 전달하는 스트림의 개수와 무관하게 언제든지 통신 노드(14)로부터 데이터를 수신하기 위한 단지 하나의 요청만이 기다리도록(outstanding)한다.

전술한 바와 같이, 대기시간의 감소는 매번 단계에서의 저스트-인-타임 스케줄링 알고리즘에 의존한다. 그 기본적인 원리는 하나의 스트림을 위한 데이터 흐름의 매 단계마다, 데이터는 그 데이터에 대한 요청이 도착된 때(이미) 이용 가능하다는 것이다. 이는 데이터 요청을 보내고 임의의 데이터 전송을 수행하는데 필요한 대기 시간을 감소시킨다. 그래서 제어 노드(18)가 저장 노드(16)에 특정 스트림을 위한 데이터 요청을 보내면, 저장 노드(16)는 그 요청에 거의 즉시 응답할 수 있다. 이 특성은 전술한 경쟁 문제를 해결하는 방법에서 중요하다.

매체 스트리머 환경에서는 데이터에 대한 액세스가 순차적(sequential)이고 하나의 스트림에 대한 데이터 속도는 예측가능하므로, 저장 노드(16)는 특정 스트림에 대한 다음 데이터 요청이 예상되는 시기를 예측할 수 있다. 요청에 응답하여 제공될 데이터의 신원(identity)도 또한 알려진다. 저장 노드(16)는 또한 데이터가 저장될 위치와, 다른 스트림들에 대한 예상 요청들에 대해서도 알고 있다. 상기 정보와 함께 디스크로부터의 판독 요청을 처리하는데 소요되는 예상 시간이 주어지면, 저장 노드(16)는 통신 노드(14)로부터 요청이 도착하기 직전에 데이터를 이용할 수 있도록 판독 동작(read operation)을 스케줄링한다. 예를 들어, 스트림 데이터 속도가 250KB/초이고 저장 노드(16)는 비디오의 매 4번째 마다의 세그먼트(every 4th segment)를 포함한다면, 그 스트림을 위한 데이터의 요청은 매 4초마다 도착한다. 판독 요청을 처리하는데 걸리는 시간이 500msec라면(판독 요청이 500 msec내에 끝날 것이라는 필수적인 확신을 가지고), 그 요청은 통신 노드(14)로부터의 요청에 대한 예상 수신 시점보다 최소한 500msec전에 스케줄링된다.

[C1. 제어 노드(18)의 기능들(control node functions)]

제어 노드(18)의 기능은 제어 흐름을 위해 매체 스트리머(10)와 외부세계 사이의 인터페이스를 제공하는 것이다. 제어노드는 또한 매체 스트리머(10) 자체가 분산형 시스템(distributed system)으로 구현된 경우이더라도 외부세계에는 하나의 시스템 이미지(single system image)를 제공한다. 제어 노드의 기능들은 정의된 응용 프로그램 인터페이스(Application Program Interface)(API)에 의해 구현된다. API는 비디오 데이터의 재생/녹화(playing/recording)와 같은 실시간 기능들 뿐 아니라 매체 스트리머(10)내에 비디오 내용을 생성하는 기능도 제공한다. 제어 노드(18)는 비디오를 재생 또는 정지시키기 위한 실시간 요청들을 통신 노드(14)들로 전송(forward)한다.

[C2. 통신 노드(14)]

통신 노드(14)는 실시간 비디오 인터페이스에 전용(dedicate)되는 다음과 같은 스레드(thread)(동일 프로세스내에서)들을 가진다: 접속 및 접속해제(connect/disconnect)요청 처리를 위한 스레드, 재생 및 정지(play/stop)와 잠시멈춤 및 재개(pause/resume) 요청 처리를 위한 스레드, 점프(jump)(전진 탐색 및 후진 탐색) 요청 처리를 위한 스레드. 또한 통신 노드는 저장 노드(16)로부터 나오는 스트림의 데이터를 판독하는 입력 스레드(input thread)와 데이터를 출력 포트에 기록하는 출력 스레드(output thread)를 가진다.

비디오를 재생하는 동안의 데이터 처리를 위한, 통신 노드(14)내 데이터 흐름 구조가 제3도에 도시된다. 이 데이터 흐름 구조는 저장 노드(16)로부터 데이터를 얻는 입력 스레드(100)를 포함한다. 입력 스레드(100)는 저장 노드들로부터의 데이터 수신을 직렬화시켜(serialize), 한번에는 단 하나의 저장 노드만이 데이터를 전송할 수 있게 한다. 출력 스레드(102)가 버퍼로부터 스트림으로 기록할 필요가 있는 경우, 입력 스레드(100)는 이 버퍼가 이미 데이터로 꽉차 있음을 보증한다. 또한, 스트림들의 입력 및 출력 동작 모두를 스케줄링하는 스케줄러 기능(104)도 존재한다. 이 기능은 상기 입력 스레드(100) 및 출력 스레드(102) 모두에 의해 사용된다.

각 스레드는 요청들의 대기행렬(queue of request)을 처리한다. 출력 스레드(102)를 위한 요청 큐(request queue)(106)는, 스트림을 식별하고 관련 버퍼의 비워져야 할 필요가 있는 시점(point)들을 식별하는 요청을 포함한다. 이 요청들은 그들이 비디오 출력 인터페이스에 기록될 필요가 있는 시간의 순서대로 정렬된다. 출력 스레드(102)가 버퍼를 비우면, 출력 스레드는 버퍼가 비었음을 표시하고 스케줄러 기능(104)을 호출하여 입력 큐(108)내에 있는 그 스트림에 대한 요청을 입력 스레드에 대기(queue)시킨다(버퍼를 채우기 위해). 입력 스레드(100)를 위한 큐

(108)도 또한 버퍼가 채워질 필요가 있는 시간 순서대로 정렬된다.

입력 스투드(100)는 또는 요청시간에 따라 정렬된 요청 큐(request queue)(108)를 처리한다(works off). 그의 임무는 저장 노드(16)로부터 버퍼를 채우는 것이다. 상기 큐내에 있는 각 요청마다, 입력 스투드(100)는 다음과 같은 동작을 취한다. 입력 스투드(100)는 스트림에 대한 다음 데이터 세그먼트를 가진 저장 노드(16)를 판별한다(바람직하게는 하나의 비디오 스트림을 위한 데이터가 다수의 저장 노드들에 걸쳐 분할되어(striped)있다). 그후 입력 스투드(100)는 앞에서 판별된 저장 노드에게 그 스트림에 대한 데이터 요청을 보내고(스위치(12)를 통과하는 메시지를 사용하여), 그 데이터가 도착하기를 기다린다. 이 프로토콜은 어느 시점에서나 특정 통신 노드(14)에는 단지 하나의 저장 노드(16)만이 데이터를 전송함을 보장한다. 즉, 이 프로토콜은 저장 노드들이 통신 노드(14)에 비동기적으로(asynchronously) 데이터를 보내려고 하는 경우에 발생할 수도 있는 충돌을 제거한다. 요청된 데이터가 저장 노드(16)로부터 수신되면 입력 스투드(100)는 버퍼가 가득 차음을 표시하고, 스케줄러(104)로 하여금 그 버퍼를 비우기 위한 요청을 출력 스투드(102)에 버퍼링하도록(스트림의 데이터 속도에 기초하여)한다.

[C3. 저장 노드(16)]

스트림의 재생을 지원하는 데이터 흐름을 위한 저장노드(16)의 구조가 제4도에 도시된다. 저장 노드(6)는 비디오 데이터를 포함하는 버퍼들의 풀(a pool of buffers)을 구비한다. 저장 노드는 각 논리적 디스크 드라이브마다 입력 스투드(110)를 가지며 또한 스위치 매트릭스(switch matrix)(12)를 통해 통신 노드밖으로 데이터를 기록하는 출력 스투드(112)를 가진다. 저장 노드는 또한 동작들의 스케줄을 위해 입력 및 출력 스투드들(110,112)에 의해 사용되는 스케줄러 기능(114)을 갖는다. 또한 데이터를 요구하는 통신 노드(14)로부터의 요청들을 처리하는 메시지 스투드(116)를 갖는다.

데이터를 요청하는 통신 노드(14)로부터 메시지가 수신되면, 메시지 스투드(116)는 대개 요청된 데이터가 이미 버퍼된 것을 발견하고 그 요청을 출력 스투드에 대기시킨다(큐118). 상기 요청들은 시간 순서대로 대기된다. 출력 스투드(112)는 상기 버퍼를 비우고 이 버퍼를 프리 버퍼들의 목록(a list of free buffers)에 추가한다. 각 입력 스투드(110)들 마다 자신의 요청 큐를 갖는다. 자신의 관련 디스크 드라이브상에 비디오 데이터를 보유한 활동 스트림(active stream)들마다, 다음 버퍼를 채우기 위해 요청시간대로(데이터 속도, 스트라이프화 수준(striping level)등에 기초하여) 정렬된 하나의 큐(120)가 유지된다. 입력 스투드는 큐(120)에 있는 첫 번째 요청을 가져와서, 그 요청에 하나의 프리 버퍼를 연관시키고, 상기 버퍼를 디스크 드라이브로부터의 데이터로 채우기 위해 I/O 요청을 발행한다. 버퍼가 가득차면, 그 버퍼는 가득찬 버퍼들의 목록(list of full buffers)에 추가된다. 이 목록은 스트림을 위한 데이터 요청을 받았을 때 메시지 스투드(116)에 의해 검사되는 목록이다. 데이터를 위한 메시지가 통신 노드(14)로부터 수신되고 필요한 버퍼가 가득차지 않았다면, 이는 최종기한을 놓친 것으로(missed deadline) 간주된다.

[C4. 저스트-인-타임 스케줄링]

저스트-인-타임 스케줄링(just-in-time-scheduling)기법은 통신 노드(14) 및 저장 노드(16) 모두에서 사용된다. 이 기법은 다음과 같은 파라미터들을 이용한다:

bc = 통신 노드(14)의 버퍼 크기

bs = 저장 노드(16)의 버퍼 크기

r = 비디오 스트림 데이터 속도

n = 비디오 스트림을 위한 데이터를 포함하는 비디오 스트라이프들의 개수

sr = 스트라이프 데이터 속도

$sr = r/n$

사용하는 알고리즘은 다음과 같다:

(1) sfc = 통신 노드에서 (어떤) 스트림의 요청 빈도수 = r/bc

(2) dfc = 저장 노드에서의 디스크 판독 요청 빈도수 = sr/bs

비디오 데이터의 스트라이프화(striping)는 뒤따르는 섹션H에서 상술할 것이다.

요청들은 상기 수식에 의해 결정된 빈도수(frequency)에 따라 스케줄링되며, 그 데이터가 필요하기 전에 그 요청

들이 미리 완료되도록 스케줄링 된다. 이는 비디오 스트림을 재생하기 시작할 때, 데이터 파이프(data pipe)에 데이터를 채움(priming)"으로써 달성된다.

sfc와 dfc의 계산은, 스트림을 재생하는 통신 노드(14) 및 비디오 데이터를 포함하는 저장 노드(16)와의 접속시간(connect time)에 행해진다. 저장 노드(16)에서 디스크로부터 입력을 스케줄링하고(제4도 참조), 통신 노드(14)에서 포트로의 출력을(그리고 저장 노드들로 부터의 입력을) 스케줄링하는데(제3도 참조) 상기 빈도수(또는 그 역인 시구간(interval)가 사용된다.

저스트-인-타임 스케줄링의 예:

4개의 저장 노드들에 스트라이프화된(분할된) 비디오로부터 2.0M비트/초 (250,000바이트/초)의 속도로 스트림을 재생한다고 하자. 또한 통신 노드의 버퍼 크기는 50,000 바이트이며 디스크 노드의 버퍼 크기는 250,000 바이트라 가정하자. 또한 데이터는 250,000바이트/초로 세그먼트들에 스트라이프화 된다고 가정하자.

저스트-인-타임 알고리즘내 여러 파라미터들의 값은 다음과 같다:

$bc = 250,000$ 바이트(통신 노드(14)의 버퍼 크기)

$bs = 250,000$ 바이트(저장 노드(16)의 버퍼 크기)

$r = 250,000$ 바이트/초(스트림 데이터 속도)

$n = 4$ (스트림에 대해 비디오가 스트라이프화된 스트라이프들의 개수)

$sr = r/n = 6250$ 바이트/초 또는 250,000/4초, 즉 매 4초마다 250,000 바이트.

$sfc = r/bc = 1/초$ (통신 노드(14)에서 요청의 빈도)

$dfc = r/bc = 1/초$ (저장 노드(16)에서 요청의 빈도)

스트림의 재생을 맡고 있는 통신 노드(14)가 1/초(1초에 한번)의 빈도로 또는 1.0초 마다 입력 및 출력 요청들을 스케줄링한다. 통신 노드(14)가 그 스트림에 전용되는 버퍼를 두 개 구비한다고 가정하면, 통신 노드(14)는 비디오 스트림의 출력을 시작하기도 전에 이미 가득찬 두 개의 버퍼를 가지고 있음을 확인한다.

접속 시간에, 통신 노드(14)는 비디오 데이터의 스트라이프를 포함하는 4개 저장 노드(16) 모두에게 메시지를 보낼 것이다. 저장 노드들중 처음 두 개는 상기 스트라이프들로부터의 첫 세그먼트에 대한 요청을 예측할 것이며, 버퍼들을 채우기 위한 디스크 요청들을 스케줄링할 것이다. 통신 노드(14)는 크기가 각각 250,000 바이트인 두 개의 버퍼에 상기 첫 두 개 세그먼트들을 판독하기 위한 입력 요청들(제3도)을 스케줄링한다. 재생 요청이 오면, 통신 노드(14)는 우선 상기 두 개 버퍼가 찼는지 확인하고 모든 저장 노드(16)들에게 곧 재생이 시작될 것임을 알린다. 그 후 통신 노드(14)는 스트림을 재생하기 시작한다. 첫 번째 버퍼가 출력을 하면(2 M비트/초(또는 250,000 바이트/초) 속도로 1초 소요), 통신 노드(14)는 저장 노드(16)로부터 데이터를 요청한다. 통신 노드(14)는 각 저장 노드로부터 1초 간격으로 차례대로 데이터를 요청한다. 즉 통신 노드는 4초 간격으로 특정 저장 노드로부터 데이터를 요구할 것이다. 통신 노드는 항상 한번에 250,000 바이트의 데이터를 요구한다. 통신 노드(14)가 저장 노드(16)로부터 데이터를 요청할 빈도수의 계산은 통신 노드(14)에 의해 접속 시간에 행해진다.

저장 노드(16)는 다음과 같이 스트림 데이터에 대한 요청을 예측한다. 스트라이프 3을 포함하는 저장 노드(16)(후술하는 섹션 H 참조)는 재생이 시작된 1초 뒤, 그리고 그후 매 4초마다 그 다음 250,000 바이트 세그먼트의 요청을 예측할 수 있다.

스트라이프 4를 포함하는 저장 노드(16)는 재생이 시작된 2초 뒤, 그리고 그후 매 4초마다 요청을 예측할 수 있다. 스트라이프 2를 포함하는 저장 노드(16)는 재생이 시작된 4초 뒤, 그리고 그후 매 4초마다 요청을 예측할 수 있다. 즉, 각 저장 노드(16)는 일정 개시 시간부터 매 4초마다 250,000 바이트의 빈도로 디스크로부터의 입력을 스케줄링한다(전술한 바와 같이), 상기 스케줄링은 재생 명령을 수신한 후에, 또 그 스트림을 위한 버퍼가 출력된 후에 저장 노드(16)에서 이루어진다. 요청 빈도수의 계산은 접속 요청이 수신된 때에 행해진다.

통신 노드(14) 및 저장 노드(16)에서 서로 다른 버퍼 크기를 사용하는 것도 가능하다. 예를 들어 통신 노드(14)에서의 버퍼 크기는 50,000 바이트이고 저장 노드(16)에서의 버퍼 크기는 250,000 바이트일 수 있다. 이런 경우, 통신 노드(14)에서의 요청 빈도수는(250,000/50,000) 5/초 또는 0.2초마다인데 반하여 저장 노드(16)에서의 요청 빈도수는 1/초 그대로이다. 통신 노드(14)는 첫 번째 스트라이프를 포함하는 저장 노드로부터 처음 두 버퍼들

(100,000 바이트)을 판독한다(세그먼트 크기는 250,000 바이트이며 첫 번째 세그먼트를 포함하는 저장 노드(16)는 디스크로부터의 입력을 접속 시간에 스케줄링함에 주목하자). 재생이 시작되면 통신 노드(14)는 저장 노드(16)에게 재생 시작 사실을 알리고 제1버퍼를 출력한다. 이 버퍼가 비워지면, 통신 노드(14)는 다음 입력을 스케줄링한다. 버퍼들은 매 0.2초마다 비워지며 통신 노드(14)는 저장 노드(16)들로부터의 입력을 상기 빈도로 요청하며 또한 동일한 빈도로 출력을 스케줄링한다.

본 예에 있어, 저장 노드(16)는 0.2초 간격으로 5개의 요청들이 도착할 것을 예상한다(이미 100,000 바이트가 판독된 첫 번째 세그먼트는 제외되므로, 초기에는 3개의 요청이 재생 후 매 4초마다 도착한다. 즉 5개 요청 중의 그 다음 시퀀스는(각각 50,000 바이트인) 그 전 시퀀스의 마지막 요청 후 4초 후에 도착한다). 저장 노드에서의 버퍼 크기가 250,000 바이트이므로 저장 노드(16)는 디스크로부터의 입력을 매 4초마다 스케줄링할 것이다(전술한 바와 똑같이).

[C5. 재생 동작의 상세한 내용(details of a play action)]

다음 단계들은 한 스트림의 재생 동작을 위한 제어 흐름 및 데이터 흐름을 추적한다. 재생을 위해 비디오를 셋업(set up)하는 단계가 제5도에 도시되어 있다. 이 단계들은 시간순서(time order)대로 되어 있다.

1. 사용자는 이미 적재된 특정 비디오로 포트 하나를 셋업하는 명령을 불러낸다. 이 요청은 제어 노드(18)로 보내진다.
2. 제어 노드(18)의 스레드는 상기 요청과 VS-CONNECT 기능을 수신한다.
3. 상기 제어 노드 스레드는 상기 비디오 카탈로그의 항목(catalog entry)을 열고(open), 그 비디오를 위한 메모리 기술자(memory descriptor)를 스트라이프된 파일 정보로 셋업한다.
4. 제어 노드(18)는 하나의 통신 노드(14)를 할당하고 상기 요청에 대해 이 통신 노드의 출력 포트를 할당한다.
5. 제어 노드(18)는 할당된 상기 통신 노드(14)에 메시지를 보낸다.
6. 통신 노드(14)내의 스레드가 제어 노드로부터 상기 메시지를 수신한다.
7. 통신 노드 스레드는 스트라이프 파일들을 포함하는 저장 노드(16)에 개방 요청(open requests)을 보낸다.
- 8, 9. 상기 개방 요청이 보내진 각 저장 노드(16)내 스레드가 상기 요청을 수신하고 요청된 스트라이프 파일을 개방하며 디스크로부터의 입력을 스케줄링할 뿐 아니라 필요한 자원들을 할당한다(상기 스트라이프 파일이 처음의 몇몇 세그먼트들을 포함하는 경우).
10. 저장 노드 스레드는 응답(response)을 스트라이프 파일의 이름(handle)(식별자)과 함께 통신 노드에 되보낸다.
11. 통신 노드(14)내의 스레드는 관련된 모든 저장 노드들로부터의 응답들을 기다리며 성공적인 응답들을 받자마자, 출력 포트를 셋업하는 것을 포함하여 스트림에 대한 자원들을 할당한다.
12. 통신 노드(14)는 비디오 데이터 파이프라인을 채우기(prime) 위해 입력을 스케줄링한다.
13. 통신 노드(14)는 제어 노드(18)에 응답을 되돌려준다.
14. 제어 노드 스레드는 통신 노드(14)로부터 성공적인 응답을 받자마자 본 실시예의 스트림에 관련된 후속 요청들에서 사용될 스트림의 이름(handle for the stream)을 사용자에게 리턴(return)한다.

다음은, 하나의 비디오 스트림이 성공적으로 셋업된 후 재생 요청을 받았을 때 취해지는 동작들을 시간 순서대로

보인 단계들이다. 이 단계들은 제6도에 도시되고 있다.

1. 사용자가 재생 명령을 호출한다.
2. 제어 노드(18)의 스레드가 상기 요청을 받는다.
3. 제어 노드(18)내 스레드는 상기 요청이 셋업된 스트림에 대한 것임을 확인(verify)하고, 할당된 통신 노드(14)에 재생 요청을 보낸다.
4. 통신 노드(14)내 스레드가 상기 재생 요청을 수신한다.
5. 통신 노드(14)는 상기 재생 요청을 그와 관련된 모든 저장 노드(16)들에게 전달하여, 저장 노드들이 상기 스트림을 위한 후속 요청의 예측을 통해 그들 자신의 동작들을 스케줄링할 수 있게 한다. 여기서 관련된(involved) 저장 노드란, 관심있는 비디오 표현의 스트라이프를 최소한 한 개라도 저장하고 있는 저장 노드를 말한다.
6. 관련된 각 저장 노드(16)내의 스레드는 상기 요청을 수신하고 상기 스트림에 대해 장래에 있을 요청들을 지원하기 위한 스케줄을 셋업한다. 관련된 각 저장 노드(16)는 통신 노드(14)에 응답을 되보낸다.
7. 통신 노드 스레드는 파이프 라인이 채워져 있음(비디오 데이터로 사전 적재되었음)을 확인하고 출력을 위해 스트림을 인에이블시킨다.
8. 통신 노드(14)는 제어 노드(18)에 응답을 되보낸다.
9. 제어 노드(18)는 사용자에게 스트림이 재생되고 있다는 응답을 보낸다.

입력 및 출력 스레드들은 정지/잠시멈춤 명령이 수신되거나 비디오가 끝날 때까지, 지정된 포트에 계속 비디오 표현을 전달한다.

[D.매체 스트리머에 대한 사용자 및 응용 인터페이스(User and application interface to media streamer)]

매체 스트리머(10)는 외부의 제어 시스템으로부터 제어 명령을 받았을 때 비디오 서버 동작을 수행하는 수동 서버(passive server)이다. 제7도는 매체 스트리머(10) 응용을 위한 시스템 구성을 보이고 그 시스템내에 존재하는 인터페이스들을 도시한다. 매체 스트리머(10)는 사용자 및 응용 프로그램들에 대해 두 개의 레벨로 이루어진 인터페이스를 제공한다:

사용자 인터페이스(제7도의 (A)) 및 응용 프로그램 인터페이스(제7도의 (B)).

상기 두가지 레벨의 인터페이스는 클라이언트 제어 시스템(client control system)들에 제공되며, 이 시스템들이 원격 프로시저 호출(remote procedure call)(RPC) 메커니즘을 통해 매체 스트리머(10)와 통신한다. 매체 스트리머(10) 대신 상기 클라이언트 제어 시스템에 인터페이스들을 제공함으로써, 응용 소프트웨어를 매체 스트리머(10)로부터 분리할 수 있다.

이렇게 하면(매체 스트리머가) 클라이언트 제어 시스템상의 응용 소프트웨어를 변경 또는 대체할 것을 요구하지 않으므로, 매체 스트리머(10)의 개량(upgrading) 또는 대체(replacing)가 용이하게 된다.

[D1. 사용자의 통신(user communications)]

매체 스트리머(10)는 두가지 유형의 사용자 인터페이스를 제공한다:

명령 라인 인터페이스(command line interface) 및 그래픽 사용자 인터페이스(graphical user interface).

[D1.1. 명령 라인 인터페이스]

명령 라인 인터페이스는 사용자 콘솔(console) 또는 인터페이스(제1도의 65, 66) 상에 프롬프트(prompt)를 디스플레이 한다. 명령 프롬프트의 뒷부분에, 사용자가 명령을 입력하되 명령 키워드를 먼저 입력하고 파라미터들을 나중에 입력한다. 이 명령이 실행된 후, 인터페이스는 다시 프롬프트를 디스플레이하고 다음 명령의 입력을 기다린다. 매체 스트리머의 명령 라인 인터페이스는 특히 다음과 같은 유형의 동작들에 적합하다:

일괄 제어(Batch Control) : 일괄 제어는 일련의 비디오 제어 명령들을 포함하는 하나의 명령 대본(command script)의 실행을 수반한다. 예를 들어 방송업계에서는 사전에 녹화되고 스케줄링된 프로그램들을 장기간동안 포함하기 위해 미리 명령대본이 준비될 수 있다. 계획된 시작 시간에, 단일 일괄 명령에 의해 상기 명령 대본이 실행됨으로써 더 이상 조작자(operator)의 간섭없이 방송을 시작할 수 있다.

자동 제어(Automatic Control): 자동 제어는, 매체 스트리머(10)에 저장된 자료들을 갱신(update)/재생(play)하기 위한 프로그램에 의해 생성된 명령 리스트의 실행을 수반한다.

예를 들어, 통신사(news agency)는 매일 매체 스트리머(10)에 날마다 새로운 자료를 적재할 수 있다. 새로운 자료들을 관리하는 응용 제어 프로그램은, 매체 스트리머(10)를 새로운 자료들로 갱신하기 위한 매체 스트리머 명령들(예를 들면 적재(Load), 삭제(Delete), 언로드(unload)명령)를 생성할 수 있다. 이 생성된 명령들은 실행되기 위해 명령 라인 인터페이스로 파이프를 통해 전달될 수(piped) 있다.

[D1.2. 그래픽 사용자 인터페이스]

제8도는 매체 스트리머 그래픽 사용자 인터페이스의 한 예이다. 이 인터페이스는 비디오 카세트 녹화기(video cassette recorder)의 제어 패널(control panel)과 유사하게, 재생(play), 잠시 멈춤(pause), 되감기(rewind) 및 정지(stop)와 같은 제어 버튼을 구비한다. 또한 어떤 동작이 사용자의 선택을 수반하는 경우(예를 들어 적재 명령은 사용자로 하여금 적재될 비디오 표현을 선택할 것을 요구한다), 상기 인터페이스는 선택 패널(selection panel)을 제공한다. 그래픽 사용자 인터페이스는 사용자와 직접적인 상호작용을(direct interaction)해야하는 경우 특히 유용하다.

일괄(batch) 버튼(130)과 수입/수출(Import/Export) 버튼(132)도 상기 그래픽 사용자 인터페이스에 포함된다. 이들의 기능은 후술하기로 한다.

[D2. 사용자 기능들(user functions)]

매체 스트리머(10)는 3가지 일반적인 유형의 사용자 기능들을 제공한다.

수입/수출 기능, VCR과 유사한 재생 제어 기능 및 고급 사용자 제어 기능.

[D2.1. 수입/수출(Import/Export)]

수입/수출 기능은 매체 스트리머(10)로부터 비디오 데이터를 빼거나 매체 스트리머(10)에 비디오 데이터를 넣는데 사용된다. 비디오가 클라이언트 제어 시스템으로부터 매체 스트리머로 들어가는 경우(수입), 비디오 데이터의 소스(source)는 클라이언트 제어 시스템의 파일 또는 디바이스로 지정된다. 비디오 데이터의 타겟(target)은 매체 스트리머(10)내의 유일한 명칭(unique name)으로 지정된다. 비디오가 매체 스트리머로부터 나와서 클라이언트 제어 시스템으로 이동하는 경우(수출), 비디오 데이터의 소스는 매체 스트리머(10)내 명칭에 의해 특정되며 비디오 데이터의 타겟은 클라이언트 제어 시스템의 파일 또는 디바이스로 지정된다.

사용자 기능들중 상기 수입/수출 카테고리내에서, 매체 스트리머(10)는 또한 비디오 하나를 제거하는 삭제(delete) 기능과, 저장된 비디오들에 관한 정보(명칭, 데이터 속도와 같은)를 얻기 위한 속성 얻기(get attributes) 기능도 제공한다.

그래픽 사용자 인터페이스를 통해 수입/수출 기능들을 불러내기 위해, 사용자는 수입/수출 소프트 버튼(132)을

클릭(click)한다(제8도). 이는 상기 개별적 기능들을 불러내기 위하여 수입, 수출, 삭제, 속성 얻기를 포함하는 새로운 패널(도시되지 않음)을 보이게 한다.

[D2.2. VCR과 유사한 재생 제어(VCR-like play controls)]

매체 스트리머(10)는 VCR과 유사한 재생 제어 세트를 제공한다. 제8도에 보인 매체 스트리머 그래픽 사용자 인터페이스는 다음과 같은 기능들이 이용 가능함을 보인다: 적재(Load), 꺼냄(Eject), 재생(Paly), 천천히(Slow), 잠시멈춤(Pause), 정지(Stop), 되감기(Rewind), 고속 전진(Fast Forward), 소리안내기(Mute). 상기 기능들은 그래픽 사용자 인터페이스상의 해당 소프트 버튼들을 클릭함으로써 활성화된다. 매체 스트리머 명령 라인 인터페이스도 비슷한 기능들의 세트를 제공한다:

셋업(setup)- 비디오를 특정 출력 포트로 셋업한다.

비디오 카세트를 VCR에 적재하는 것과 유사하다.

재생(play)-셋업된 비디오를 재생하기 시작하거나 잠시 멈춰진(paused) 비디오 재생을 재개한다.

잠시멈춤(pause)-비디오의 재생을 잠시 멈춘다.

분리(detach)-VCR로부터 비디오 카세트를 꺼내는 (ejecting)것과 유사하다.

상태 표시(status)- 어떤 비디오가 재생되고 있는지, 경과된 재생시간은 얼마인지 등 포트의 상태를 표시한다.

[D2.3. 고급 사용자 제어(Advanced user controls)]

방송업계와 같은, 특정 응용의 요구사항들을 지원하기 위하여, 본 발명은 몇몇 고급 사용자 제어를 제공한다.

재생 리스트(play list)- 하나의 포트상에 재생될 다수의 비디오 및 그들의 순서(sequence)를 셋업한다.

재생 기간(play length)- 비디오가 재생될 시간을 제한한다.

일괄 동작(batch operation)- 명령 파일에 저장된 동작들의 리스트를 실행한다.

재생 리스트 및 재생 기간의 제어는 그래픽 사용자 인터페이스상의 적재 버튼(134)으로 달성된다. 각 셋업 명령은 특정 포트에 대한 재생 리스트에 추가될 하나의 비디오를 지정할 것이다. 셋업 명령은 또한 비디오가 재생될 수 있는 시간 제한(time limit)을 지정한다. 제9도는 그래픽 사용자 인터페이스상의 적재 소프트 버튼(134)을 클릭함에 응답하여 나타난 패널을 보인 것으로서 이는 상기 재생 리스트에 추가될 비디오를 선택하고 그 비디오를 재생하는 시간 제한을 지정하기 위한 것이다. 사용자가 화일들 상자(136)의 파일 이름 하나를 클릭하면 그 파일 이름이 화일 이름 상자(138)로 입력된다. 사용자가 추가(Add) 버튼(140)을 클릭하면, 화일 이름 상자(138) 안에 있는 파일 이름이 그 시간 제한과 함께 재생 리스트상자(142)에 추가되며(appended), 현재 재생 리스트를 보여준다(재생 리스트상의 각 비디오에 대한 시간 제한과 함께).

일괄 동작(batch operation)은 상기 그래픽 사용자 인터페이스의 일괄 소프트 버튼(130)을 사용하여 달성된다(제8도 참조).

일괄 버튼(130)이 활성화되면, 사용자가 명령 파일 이름을 선택 또는 입력할 수 있도록 일괄 선택 패널(batch selection panel)이 보여진다(제10도 참조). 일괄 선택 패널상의 실행(Execute) (144)을 클릭하면 선택된 명령 파일내의 명령들이 실행되기 시작한다. 제10도는 그래픽 사용자 인터페이스상의 일괄 및 실행 동작에 대한 하나의 예이다. 예를 들어서, 사용자가 우선 c:/batchcmd 디렉토리내의 batch2 파일에 명령 대본을 생성한다. 그 후 사용자는 제8도에 보인 그래픽 사용자 인터페이스상의 일괄 버튼(130)을 클릭하여 일괄 선택 패널을 보이게 한다. 그런 다음 사용자는 일괄 선택 패널의 디렉토리(directory) 상자(146)내에 있는 c:/batchcmd를 클릭한다. 그 결과

화일들 상자(148)에 파일들의 리스트가 보여진다. 화일들 상자(148)내의 batch2 라인을 클릭하면 화일 이름 상자(150)에 batch2가 입력된다. 최종적으로, batch2 파일 안에 저장된 명령들을 순서대로 실행시키기 위하여 사용자 는 실행 버튼(144)을 클릭한다.

[D3. 응용 프로그램 인터페이스]

매체 스트리머(10)는 응용 제어 프로그램들이 매체 스트리머(10)와 상호작용하고 그 동작을 제어하게 하기 위해서 (다시 제7도를 참조), 전술한 응용 프로그램 인터페이스(Application Program Interface)(API)를 제공한다. API는 원격 프로시저 호출에 기초한(RPC-based)프로시저들로 구성된다. 응용 제어 프로그램은 프로시저 호출을 함으로써 API 기능들을 실시한다. 프로시저 호출의 파라미터가 수행될 기능을 지정한다. 응용 제어 프로그램은 매체 스트리머(10)의 물리적 및 논리적 위치에 관계없이 API 기능들을 실시한다. 비디오 서비스를 제공할 매체 스트리머(10)의 신원(identity)은, 클라이언트 제어 시스템을 시작할 때(startup time)에 설정되며, 또는 응용 제어 프로그램을 개시할 때(initiation)에 설정될 수도 있다. 매체 스트리머(10)의 신원이 설정되고 나면 상기 프로시저 호출은 서비스를 위해 정확한 매체 스트리머(10)로 향한다.

뒤에서 지적하는 것을 제외하고, API 기능들은 동기적으로(synchronously) 처리된다. 즉 일단 기능 호출이 호출자(caller)에게로 복귀되고 나면 그 기능은 종료되고 매체 스트리머(10)에서는 더 이상의 추가적인 처리가 필요하지 않다. API 기능들을 동기적인 동작(synchronous operation)들로 구성함으로써, 문맥 전환(context switching), 비 동기 시그널링 및 피드백(synchronous signalling and feedback)을 위한 부가적 처리의 오버헤드를 피할 수 있다. 이러한 성능은 엄격한 실시간 요구 때문에 비디오 서버 응용에서 중요하다.

API 기능들의 처리는 요청을 받은 순서대로 이루어진다. 이는 사용자 동작들이 올바른 순서로 처리됨을 확인한다. 예를 들어, 비디오는 재생되기 전에 반드시 접속(셋업)되어야 한다. 또다른 예로서, 재생 요청을 그에 뒤따르는 잠시 멈춤 요청과 순서를 바꾼다면 사용자에게는 완전히 다른 결과를 보일 것이다.

VS-PLAY 기능은 비디오의 재생을 시작시키며 호출자(caller)에게 제어를 즉시 (비디오 재생이 끝날 때 까지 기다리지 않고) 반환한다. 이러한 구조의 이론적 근거는, 비디오 재생에 소요되는 시간은 보통 길며(몇분에서 몇시간까지) 예측할 수 없기(잠시 멈춤이나 정지 명령들이 있을 수 있다) 때문에, VS-재생 함수를 비동기적으로 만듦으로써 그렇지 않았다면 예측할 수 없는 오랜 시간동안 할당되었을 자원들을 자유롭게 하는 것이다.

비디오 재생이 종료되면 매체 스트리머(10)는 응용 제어 프로그램에게 비디오가 끝난 사실을 알리기 위해서, 응용 제어 프로그램에 의해 지정된 시스템/포트 주소로 비동기 호출을 발생한다. 상기 시스템/포트 주소는, 응용 제어 프로그램이 비디오를 연결하기 위해 API VS-CONNECT 기능을 호출할 때 응용 제어 프로그램에 의해 지정된다. VS-PLAY를 위한 콜백(callback) 시스템/포트 주소는 개별적인 비디오 레벨에서 지정된다. 이는 응용 제어 프로그램이 비디오 종료메시지를 임의의 제어점(control point)으로 향하게 할 수 있는 특권을 가졌음을 의미한다. 예를 들어, 한 응용은 다수 또는 전체 클라이언트 제어 시스템들을 위한 비디오 종료 메시지를 처리하기 위해 하나의 중앙적 시스템/포트를 사용하고자 할 수 있다. 다른 응용에서는 하나의 클라이언트 제어 시스템을 위한 비디오 종료 메시지 처리에 여러개의 서로 다른 시스템/포트 주소들이 사용될 수 있다.

상기 API 구조를 사용하면, 매체 스트리머(10)는 이종적인(heterogeneous) 하드웨어 및 소프트웨어 플랫폼(platform)으로된 다수의 병행(concurrent) 클라이언트 제어 시스템들에게, 요청된 동작들의 올바른 순서화를 확실히 하면서도 동기 및 비동기 유형의 동작들 모두를 효율적으로 처리하도록 지원할 수 있다. 예를 들어, 매체 스트리머(10)는 PS/2 시스템상에서 실행되는 IBM OS/2 운영 체제를 사용할 수 있는 반면 클라이언트 제어 시스템은 RS/6000 시스템 상에서 실행되는 IBM AIX 운영체제를 사용할 수 있다. (IBM, OS/2, PS/2, AIX 및 RS/6000은 모두 IBM 사의 등록 상표이다).

[D4. 클라이언트/매체 스트리머 통신(client/media streamer communications)]

클라이언트 제어 시스템 및 매체 스트리머(10)간의 통신은 예를 들어 기존 유형의 원격 프로시저 호출(RPC) 기능을 통해 이루어진다. 제11도는 클라이언트 제어 시스템(11)과 매체 스트리머(10) 사이의 통신을 위한 RPC 구조를 보인다. 매체 스트리머 기능들을 호출함에 있어서, 클라이언트 제어 시스템(11)은 RPC 클라이언트 역할을 하며 매체 스트리머(10)는 RPC 서버 역할을 한다. 이는 제11도의 (A)에 표시된다. 그러나 비동기적 기능 즉 VS-PLAY에 있어서는 그 종료가 매체 스트리머(10)로 하여금 클라이언트 제어 시스템으로의 호출을 발생시킨다. 이 경우, 클라이언트 제어 시스템(11)은 RPC 서버 역할을 하고 기능하고 매체 스트리머(10)는 RPC 클라이언트 역할을 한

다. 이는 제11도의 (B)에 표시된다.

[D4.1. 클라이언트 제어 시스템(11)]

클라이언트 제어 시스템에서, 사용자 명령 라인 인터페이스는 3개의 내부 병렬 프로세스(스레드)로 구성된다. 제1 프로세스는 매체 스트리머(10)로의 RPC 호출(제11도의 (A) 참조)을 하는 API 기능들을 불러냄으로써 사용자 명령 입력을 해석(parse)하고 그 요청된 동작을 수행한다. 또한 이 프로세스는 여러 출력 포트에서 셋업되고 재생되는 비디오들의 상태를 추적한다. 제2프로세스는 각 비디오의 경과된 재생 시간을 그 비디오의 특정한 시간 제한에 대해서 주기적으로 검사한다. 비디오가 그의 시간 제한에 이르면 그 비디오는 정지되고 접속이 끊어지며(disconnected), 동일한 출력 포트에 대한 대기 큐(waiting queue) (만약 존재한다면)에 있는 다음 비디오가 시작된다. 클라이언트 제어 시스템(11)의 제3프로세스는 매체 스트리머(10)로부터 VS-PLAY의 비동기 종료 통지(asynchronous termination notification)를 수신하기 위해 RPC 서버 역할을 한다(제11도내 (B)).

[D4.2. 매체 스트리머(10)]

매체 스트리머(10)가 시작되는 동안, 클라이언트 제어 시스템(들)(11) 및 매체 스트리머(10) 사이의 RPC를 지원하기 위해 두 개의 병렬 프로세스(스레드)들이 호출된다. 제1 프로세스는 클라이언트 제어 시스템으로부터의 API 기능 호출을 위해 RPC 서버 역할을 한다(제11도내 (A).) 상기 제1 프로세스는 RPC 호출을 수신하며, 요청된 기능(VS-CONNECT, VS-PLAY, VS-DISCONNECT) 수행을 위해 적절한 프로시저들을 디스패치(dispatch)한다. 제2 프로세스는 응용 제어 프로그램에게 비동기 종료 사실을 알리기 위해서 적절한 클라이언트 제어 시스템 주소를 호출하는 RPC 클라이언트 역할을 한다. 제2 프로세스는, 비디오들의 재생을 처리하는 다른 프로세스들에 의해 기록되는 내부 파이프(internal pipe)에서 대기하면서 자신을 블러킹시킨다. 후자(비디오 재생을 다루는 다른 프로세스)가 비디오의 끝부분에 이르거나 비정상적인 종료 상태에 이르면, 상기 파이프에 메시지를 기록한다. 블러킹된 프로세스는 이 메시지를 읽고 적절한 클라이언트 제어 시스템(11) 포트 주소에 RPC 호출(제11도내(B))을 하여, 클라이언트 제어 시스템으로 하여금 자신의 그 상태를 갱신하고 그 상태에 알맞은 동작을 취할 수 있게 한다.

[E. 비디오 전달을 위한 매체 스트리머 메모리 구조 및 최적화 (media streamer memory organization and optimization for video delivery)]

본 발명의 일면은 비디오 전달 환경에 캐쉬 관리(cache management) 및 관련된 I/O 동작들을 적합시키기(tailoring) 위한 통합된 메카니즘을 제공하는 것이다. 본 발명의 이런 면은 이하 자세히 서술될 것이다.

[E1. 종래기술상의 캐쉬 관리(prior art cache management)]

캐쉬관리를 위한 종래의 메카니즘은 캐쉬 제어기(cache controller) 및 운영체제의 파일 서브시스템(file subsystem)에 내장된다. 이들은 범용(general purpose use)으로 설계되었으며 비디오 전달의 필요를 충족시키도록 특화되지(specialized)않았다. 제12도는 통상의 캐쉬 관리 메카니즘이 비디오 전달을 위해 구성되는 하나의 가능한 방법을 도시한다. 이 기법은 두 개의 디스크 파일들(160,162) 사이에 비디오 스플릿(video split)을 사용(하나의 파일에 저장하기에는 너무 크기 때문에)하며, 파일 시스템(164), 매체 서버(168) 및 비디오 드라이버(170)를 포함한 프로세서를 이용한다. 제12도에는 또한 두 개 비디오 스트림을 위한 두 개의 비디오 어댑터 포트(video adapter port)(172,174)들이 도시되었다. 또한 디스크 파일(160)의 세그먼트를 주기억 장치로 판독하고 그 후 그 데이터를 제1비디오 포트(172)에 기록하는 데이터 흐름과, 동일한 세그먼트를 판독하고 이를 제2비디오 포트(174)에 기록하는 데이터 흐름도 도시되어 있다. 제12도는 본 발명의 매체 스트리머(10)에 의해 언급되고 극복되는, 종래기술상 발생하는 문제점들을 보이기 위해 사용된 것이다.

[제12도의 A1내지 A2단계에 대한 설명]

A1. 매체 서버(168)가 세그먼트 sk를 비디오 드라이버(170)내 버퍼로 판독하기 위해 파일 시스템(166)을 호출한다.

A2. 파일 시스템(166)이 sk의 일부를 파일 시스템(166)내 캐쉬버퍼(cache buffer)로 읽어온다.

A3. 파일 시스템(166)이 캐쉬 버퍼를 비디오 드라이버(170)내 버퍼로 복사한다.

단계 A2 및 A3는 여러 번 반복된다.

A4. 파일 시스템(166)이 sk를 비디오 포트 1(176)에 기록하기 위해 비디오 드라이버(170)를 호출한다.

A5. 비디오 드라이버(170)가 sk의 일부분을 비디오 드라이버(170)내 버퍼로 복사한다.

A6. 비디오 드라이버(170)가 상기 버퍼를 비디오 포트 1(176)에 기록한다.

단계 A5 및 A6는 여러 번 반복된다.

단계A7 내지 A12는 포트 1이 포트2로 바뀌는 점만 제외하고는 위와 동일한 방식으로 동작한다. 만약 sk의 일부가 포트 2에서 필요할 때 파일 시스템(166)의 캐쉬안에 있다면 단계A8은 생략(skip)할 수 있다.

비디오의 전달은 다중 데이터 스트림을 통한 다량의 데이터 전송을 수반한다. 전체적인 사용 패턴은 캐쉬를 최적화하는데 사용되는 두 개의 전통적 패턴, 즉 랜덤(random) 및 순차(sequential)패턴 어느것에도 맞지 않는다. 만약 랜덤 옵션(random option)이 선택되면, 대부분의 캐쉬 버퍼들은 아마 최근에 판독된 비디오 세그먼트로 부터의 데이터를 포함하겠지만 데이터들이 배출(expire)되기 전에 그들을 판독하기 위해 기다리는 비디오 스트림은 없을 것이다. 순차 옵션(sequential option)이 선택 된다면 가장 최근에 사용된 캐쉬 버퍼들이 먼저 재사용되어, 파일 시스템 캐쉬안에서 필요한 세그먼트 부분을 발견하기는 더욱 어려워진다. 전술한 바와 같이, 비디오 전달의 중요한 요소는 데이터 스트림이 등시적으로, 즉 관람자 또는 사용자가 불만스러워할 단절(break)이나 중단(interruption)이 없이 전달되어야 한다는 것이다. 위에 보인 바와 같이 종래의 캐쉬 메카니즘(caching mechanism)은 사용자에게 대한 비디오 데이터의 등시적 전달을 보증할 수 없다.

제12도에 나타난 추가적인 문제점들은 다음과 같다:

- a. 디스크 및 비디오 포트/I/O는 일반적인 파일 시스템 요건을 만족시키기 위해 상대적으로 작은 세그먼트로 행해진다. 이는 비디오 세그먼트 크기의 세그먼트들에서 요구되는 것보다 더 많은 처리시간, 디스크 탐색 오버헤드 및 버스 오버헤드를 요구한다.
- b. 파일 시스템 캐쉬 버퍼와 매체 서버 버퍼 사이 및 매체 서버 버퍼와 비디오 드라이버 버퍼 사이의 데이터 복사를 위한 처리시간은 바람직하지 않은 오버헤드이므로 제거되는 것이 바람직하다.
- c. 동일한 비디오 세그먼트의 사본을 동시에 포함하기 위해 두 개의 비디오 버퍼들(즉 172,174)을 사용하는 것은 주 메모리를 비효율적으로 사용하는 것이다. 동일한 데이터가 파일 시스템 캐쉬 및 비디오 드라이버 버퍼에 저장되어 있다면 더욱 주 메모리를 낭비하는 것이 된다.

[E2. 비디오에 최적화된 캐쉬 관리(video-optimized cache management)]

본 발명의 상기 일면에 따르면 캐쉬 관리 동작에는 3가지 주요한 면이 존재한다: 스트림들간에 세그먼트 크기의 캐쉬 버퍼들을 공유하는 것, 예측 캐쉬, 캐쉬를 최적화시키기 위한 동기화가 그것이다.

[E2.1. 스트림들간에 세그먼트 크기의 캐쉬 버퍼 공유(sharing segment size cache buffers across streams)]

비디오들은 고정된 크기의 세그먼트(fixed size segment)로 저장되고 관리된다. 세그먼트들은 순차적으로 번호가 매겨져서, 예를 들어 세그먼트 5는 세그먼트 6보다 비디오 표현의 시작부에 더 가까운 비디오 표현부분을 저장한다. 세그먼트의 크기는 디스크 I/O, 비디오 I/O, 버스 사용 및 프로세서 사용을 최적화하도록 선택된다. 비디오의 한 세그먼트는 단지 비디오 이름과 세그먼트 번호에 따라서 고정된 내용(fixed content)을 갖는다. 디스크 및 비디오 출력으로의 모든 I/O와 모든 캐쉬 동작들은 세그먼트 경계에 정렬되어(aligned)행해진다.

본 발명의 이러한 면은, 기초를 이루는 하드웨어(underlying hardware)가 디스크 및 통신 노드(14)내 비디오 출력 카드 사이에서 통신 노드의 캐쉬 메모리를 통과하지 않고도 직접적인 데이터 흐름으로 피어-투-피어(peer-to-peer)동작을 지원하는지 여부에 따라 두가지 형태를 취한다. 피어-투-피어 동작을 위해, 캐쉬는 디스크 저장 장치 유닛(16)에서 행해진다. 피어-투-피어 동작을 지원하지 않는 하드웨어에 대해서는, 세그먼트 크기의 블록들로서

페이지 정렬되고(page-aligned)연속적인(contiguous) 캐시 메모리(통신 노드(14)내에 있는) 로데이터가 직접 판독되어 I/O 동작과 데이터 이동을 최적화한다. (후술하는 섹션 F. 비디오에 최적화된 디지털 메모리 할당 참조).

비디오 세그먼트가 더 이상 필요하지 않을 때까지 데이터는 동일한 위치에 계속 존재하며 상기 위치로부터 직접 기록된다. 비디오 세그먼트가 캐시되는 동안, 그 비디오 세그먼트를 출력할 필요가 있는 모든 비디오 스트림들은 동일한 캐시 버퍼를 액세스한다. 그러므로, 비디오 세그먼트의 사본 하나가 많은 사용자에게 의해 이용될 수 있으며, 동일한 비디오 세그먼트의 추가적 사본을 판독하기 위한 추가적인 I/O 프로세서, 버퍼 메모리 사용을 피할 수 있다. 피어-투-피어 동작을 위해서는, 남아있는 I/O의 절반 및 대부분의 프로세서와 주 메모리 사용을 통신 노드(14)에서 회피할 수 있다.

제13도는 피어-투-피어 동작이 없는 시스템을 위한 본 발명의 일 실시예를 도시한다. 비디오 데이터는, 홀수 세그먼트들이 제1디스크 저장 노드(180)에 짝수 세그먼트들이 제2디스크 저장 노드(182)에 있도록 디스크 저장 노드(16)상에 스트라이프화된다(후술하는 섹션 H 참조).

이러한 구성을 위한 데이터 흐름이 제13도에 또한 도시된다. 도면을 보면 알 수 있듯이, 세그먼트 sk가 디스크(182)로부터 통신 노드(186)내 캐시 버퍼(184)로 판독되며, 그 후 비디오 출력 포트 1 및 출력 포트 2에 기록된다. sk 비디오 데이터 세그먼트는 하나의 I/O 동작으로 직접 캐시 버퍼(184)에 판독되며 그 후 포트 1에 기록된다. 그 다음 sk 비디오 데이터 세그먼트가 하나의 I/O 동작으로 캐시 버퍼(184)로부터 포트 2로 기록되어진다.

제12도의 종래 기법에 대해 전술한 모든 문제점들은 제13도에 도시된 시스템에 의해 극복될 수 있다.

제14도는 디스크 저장 노드 및 비디오 출력 카드 사이의 피어-투-피어 동작 지원을 포함하는 구성을 위한 데이터 흐름을 도시한다. 한 쌍의 디스크 드라이브(190,192)가 스트라이프화된 비디오 표현을 포함하며 이 비디오 표현들은 중간에 긴 통신 노드(14)의 주 메모리를 통과하지 않고 직접 한 쌍의 비디오 포트들(194,196)로 입력된다.

이런 구성을 위한 데이터 흐름은 세그먼트 sk로 하여금 디스크(192)로부터 디스크 캐시 버퍼(198)를 거쳐 직접 포트 1로 판독(한번의 I/O 동작으로)되도록 하는 것이다.

만약 세그먼트 sk를 포트 2에 판독하라는 호출이 뒤따른다면, 세그먼트 sk는 디스크 캐시 버퍼(198)로부터 포트 2로 직접 판독된다(한번의 I/O 동작으로).

포트 1을 위해 디스크 캐시 버퍼(198)로 판독된 데이터가 포트 2로의 기록을 위해 계속 존재한다면, 메모리, 버스 및 프로세스 자원을 가능한 최상의 사용 결과 비디오 세그먼트가 포트 1 및 포트 2로 전송된다.

피어-투-피어와 주 메모리 캐시 메커니즘을 결합하는 것이 가능한데, 예를 들면 통신 노드(14)의 단지 한 개 포트에서만 재생되는 비디오 표현을 위해서는 피어-투-피어 동작을 사용하고 통신 노드(14)의 다수 포트들에서 재생되는 비디오 표현들을 위해서는 통신 노드(14)내에서 캐시하는 것이다.

캐시의 책임을 디스크 저장 노드와 통신 노드에 분할하는 정책은, 주어진 하드웨어 구성으로 지원될 수 있는 비디오 스트림들의 수를 최대화하기 위해서 선택된다. 만약 지원되어야할 스트림들의 개수를 안다면, 캐시 저장 장소의 크기(amount) 및 배치(placement)도 결정될 수 있다.

[E2. 2. 예측 캐시(predictive caching)]

예측 캐시 메커니즘은 비디오 전달에 잘맞는 캐시정책에 대한 필요성을 만족시킨다. 비디오 표현은 일반적으로 매우 예측하기 쉽다. 통상적으로, 비디오 표현들은 첫부분부터 재생을 시작하여 상당히 긴 소정의 시간동안 고정된 속도로 재생되고 끝부분에 이르는 경우만 정지한다. 매체 스트리머 (10)의 캐시 기법은 이러한 예측가능성(predictability)을 이용하여, 언제든지, 한번에 캐시되는 비디오 세그먼트들의 세트를 최적화한다.

상기 예측가능성은 캐시 버퍼를 채우기 위한 판독 동작을 스케줄링하는데 사용되며, 캐시 버퍼들을 재사용(reclaiming)하는 알고리즘을 구동하는데에도 사용된다. 버퍼의 내용이 소멸하기 전에 사용될 것이 예상되지 않는 버퍼들은 즉각 재사용되어 좀더 높은 우선순위의 사용을 위해 그 공간을 비워준다. 버퍼의 내용이 적당한 시간내

에 사용되기 위해 기다리는 버퍼들은 그 버퍼의 마지막 사용이 오래전에 있었다 하더라도 재사용되지 않는다.

특히, 비디오들 v_1, v_2, \dots 과 이 비디오들을 재생하는 스트림들 s_1, s_2, \dots 이 주어지면, 각 스트림 s_j 는 하나의 비디오 $v(s_j)$ 를 재생하며 $v(s_j)$ 의 k 번째 세그먼트를 기록하기 위한 예측 시간은 선형함수 $t(s_j, k) = a(s_j) + r(s_j)k$ 이며, 여기서 $a(s_j)$ 는 시작 시간 및 시작 세그먼트 번호에 따라 다르며, $r(s_j)$ 는 세그먼트 하나를 재생하는 데 소요되는 일정한 시간이고, $t(s_j, k)$ 는 스트림 s_j 의 k 번째 세그먼트를 재생하도록 스케줄된 시간이다.

이 정보는 캐쉬 버퍼를 채우기 위한 판독 동작을 스케줄링하는 데와, 캐쉬 버퍼 재사용(re-using)을 위한 알고리즘을 구동하는데 사용한다. 캐쉬 관리 알고리즘의 동작에 대한 몇몇 예는 다음과 같다:

[예 A]

현재 재생되는 비디오 스트림들중 어떤 것에 의해서도 재생될 것이 예상되지 않는 비디오 세그먼트를 포함하는 캐쉬 버퍼는, 재생될 것이 예상되는 임의의 버퍼들이 재사용되기 전에 재사용된다. 이러한 제약조건(constraint)을 만족한 후에는, 그 비디오 세그먼트를 캐쉬된 상태로 유지하는 우선순위를 결정하기 위한 가중치(weight)로서 그 비디오의 재생빈도 및 세그먼트 번호가 사용된다. 이 그룹내에서 가장 높은 유지 우선순위 (highest retention priority)는 자주 재생되는 비디오의 처음에 나오는 비디오 세그먼트에 할당된다.

[예 B]

재생될 것이 예상되는 비디오 세그먼트를 포함하는 캐쉬버퍼를 위해서는, 비디오 세그먼트를 캐쉬된 상태로 유지하는 우선순위 결정을 위한 가중치로서 다음번의 예상 재생 시간과 비디오 세그먼트 재생을 위해 남겨진 스트림들의 개수가 사용된다. 이 가중치는 필수적으로 캐쉬 버퍼의 유지 우선순위가, 캐쉬 버퍼가 재사용될 때 예상되는 I/O들 개수와(임의의 비디오 세그먼트에 대하여) 캐쉬 버퍼가 유지되는 경우 예상되는 I/O들 개수와의 차이로 설정(set)될 수 있게 한다.

예를 들어 만약 v_5 가 s_7 에서 재생되며, v_8 은 s_2 및 s_3 에서 재생되고 여기서 s_2 는 s_3 보다 5초 뒤에서 흐르며 (running), v_4 는 s_{12} 내지 s_{20} 스트림들에서 재생되며 각 스트림은 다음 스트림보다 30초 뒤에 있다면: 이미 s_7 에 의해 사용된 v_5 데이터를 포함하는 버퍼들이 가장 먼저 재사용되며, 이미 s_2 에 의해 사용된 v_8 데이터를 포함하는 버퍼들이 그 다음에 재사용되고, 이미 s_{12} 에 의해 사용된 v_4 데이터를 포함하는 버퍼들이 그 다음에 재사용되며, 가장 낮은 유지 우선순위를 가진 잔존 버퍼들이 그 다음에 재사용된다.

상기 캐쉬 관리 알고리즘은, 특별한 경우 즉 접속 동작(connection operation) (조만간 비디오 세그먼트가 재생될 것이라고 예측할 수 있지만, 정확히 언제 재생될 것인지는 예측할 수 없는 경우) 및 정지 동작(stop operation)(예전의 예측이 수정되어야 할 경우)과 같은 경우를 위한 변형들(variations)을 제공한다.

[E2.3. 캐쉬를 최적화하기 위한 스트림들의 동기화(synchronizing streams to optimize caching)]

주어진 비디오 세그먼트를 요구하는 모든 스트림들을 클러스터화(cluster)하여, 그 세그먼트를 포함하는 캐쉬 버퍼가 기억 장소내에 남아있어야 하는 시간을 최소화하고 그럼으로써 다른 비디오 스트림들을 위해 더 많은 시스템 용량을 이용가능하게 남겨두는 것도 바람직하다. 비디오의 재생 동안에, 각 세그먼트들이 재생되는 속도면에서는 보통 융통성이 없다. 그러나 어떤 비디오 전달 응용에서는 재생 속도가 융통성이 있다.(flexible). (즉, 시청자의 반발(adverse human reaction)을 야기하지 않고도 비디오 및 음성이 약간씩 빨라지거나 늦어질 수 있다). 게다가 비디오는 즉각적인 시청이 아닌 목적으로 전달될 수 있다. 만약 속도가 변동될 수 있다면, (시간적으로) 앞에 있는 스트림들은 허용가능한 최소 속도로 재생되고(시간적으로) 뒤에 있는 스트림들은 허용가능한 최대 속도로 재생되며, 스트림들 간의 갭(gap)을 메꾸고 세그먼트들이 버퍼링된 채로 있어야만 하는 시간을 감소시킨다.

동일한 비디오 표현을 사용하는 스트림들을 클러스터화 하는 것은 또한 접속 동작 및 재생 동작중에도 고려된다. 예를 들어, VS-PLAY-AT-SIGNAL은 한 개 비디오를 다수의 스트림들에 동시에 재생하기 시작할 때 사용될 수 있다. 이는 더 많은 시스템 자원들을 다른 비디오 스트림을 위해 남겨주고, 시스템의 유효 용량을 증가시켜 클러스터화를 개선한다. 특히 클러스터화는 하나의 스트림이 다른 스트림과 시간적으로 일치하도록 짧은 기간동안 하나의 스트림을 지연시킴으로써, 캐쉬내 세그먼트들의 한 사본이 두 스트림을 모두에 사용될 수 있게 하여 처리 자원 (processing assets)들을 절약한다.

[F. 비디오에 최적화된 디지털 메모리 할당(video optimized digital memory allocation)]

디지털 비디오 데이터는 통상적인 데이터 처리를 하는 데이터와 달리 랜덤하지 않고(non-random), 순차적이며(sequential), 대량이며(large), 내용이 중요하기 보다는 시간이 중요하다는(time critical rather than content critical) 속성을 가진다. 데이터의 다중 스트림들은 반드시 높은 비트 속도로 전달되어야 하므로, 데이터 경로상에서 필수적이지 않은 오버헤드는 모두 최소화될 것을 요구한다. 매체 스트리머(10)의 효율 및 용량을 최대화하기 위해서는 신중한 버퍼 관리가 요구된다. 메모리의 할당, 할당 해제 및 액세스는 이 과정에서 중요한 요소이며, 이들을 부적절하게 사용하면 메모리의 단편화(fragmentation), 효율 감소, 비디오 데이터가 지연 및 손상되는 결과가 생길 수 있다.

본 발명의 매체 스트리머(10)는 하이 레벨 응용들(high level applications)로 하여금 디지털 비디오 데이터를 위해 교체 불가능하고(non-swappable), 페이지로 정렬되어 있으며(page aligned), 연속적인 메모리 세그먼트(블럭)들을 할당하거나 할당 해제할 수 있게 하는 메모리 할당 프로시저(memory allocation procedure)를 사용한다. 이 프로시저는 비디오 전송 응용들에게는 단순하고 하이 레벨인 인터페이스를 제공하며, 요청된 크기대로 메모리 블록들을 할당하기 위해 로우 레벨 운영체제 모듈 및 코드 세그먼트들을 이용한다. 메모리 블록들은 물리적 메모리 내에서 연속되고 고정되어 있어서, 가상 메모리에서의 스와핑 또는 페이징(virtual memory swapping or paging)에서 발생할 수 있는 지연 및 손상을 없애고, 데이터 전송 소프트웨어에서 집합 및 분산 루틴(gather/scatter routines)을 구현해야 하는 복잡함을 제거한다.

하이 레벨 인터페이스는 또한 요청된 메모리 블록에 대해 다양한 주소지정 모드(addressing mode) 값을 반환하여, 하나의 매체 스트리머 환경에서 병렬로 동작하는(operating concurrently) 다양한 메모리 모델들에 맞추기 위해 값비싼 동적 주소 변환(dynamic address conversion)을 할 필요가 없다. 다양한 응용들에 의해 사용되는, 프로세스에 선형이며 프로세스에 대해 세그먼트화된 주소들(process linear and process segmented address) 뿐 아니라 고정 디스크 디바이스와 같은 기타 디바이스 드라이버들에 의해 직접 액세스하기 위한 물리적 주소가 사용 가능하다. 시스템에 메모리 블록을 반환하는 할당 해제 루틴(deallocation routine)도 제공되어, 메모리가 모두 단일 블록으로 반환되는 덕분에 단편화(fragmentation)의 문제점을 없앨 수 있다.

[F1. 메모리 할당에 사용되는 명령들]

1. 물리적 메모리 할당(Allocate Physical Memory): 요청된 크기의 메모리 블록을 할당하여, 그 블록의 길이와 그 메모리 영역의 여러 메모리 모델 주소들과 함께 하나의 제어 블록이 반환된다.
2. 물리적 메모리의 할당해제(Deallocate Physical Memory): 메모리 블록을 운영체제에 반환시키고 그에 관련된 메모리 포인터들을 자유롭게 한다.

[F2. 응용 프로그램 인터페이스]

디바이스 드라이버는 시스템 컨피규레이션 파일(system configuration file)안에 정의되며 시스템이 시작될 때 자동적으로 초기화된다. 그러면 응용이 그 디바이스 드라이버를 하나의 의사 장치(pseudo-device)로서 개방하여 그의 레이블(label)을 얻고, 그 후 인터페이스를 사용하여 명령 및 파라미터들을 전달한다. 지원되는 명령들은 메모리 할당 명령(Allocate Memory) 및 메모리 할당 해제 명령(Deallocate Memory)이며, 파라미터들은 메모리 크기 및 논리적 메모리 주소에 대한 포인터들이다. 일단 메모리의 물리적 블록이 할당되면 상기 주소들이 디바이스 드라이버에 의해 설정되며, 물리적 주소가 논리적 주소로 변환된다. 만약 상기 할당이 실패하는 경우는 널(null)값이 반환된다.

제15도는 상기 프로시저를 사용할 수 있는 전형적인 응용들의 세트를 보인다. 버퍼 1은, 변경되고 버퍼 2에 배치될 데이터를 위한 32 비트 응용에 의해 요청된다. 그 후 상기 버퍼는 세그먼트화된 주소를 사용하는 16 비트 응용에 의해 또는 고정 디스크 드라이브와 같은 물리적 디바이스에 의해 직접적으로 조작될 수 있다. 고정적이며 물리적이고 연속적인 버퍼들을 사전 할당(preallocate)하기 위해 상기 할당 스킴(allocation scheme)을 사용함으로써, 각 응용은 데이터 액세스를 위해 그의 고유한 직접 주소지정(native direct addressing)을 사용할 수 있게 되어, 주소 변환 및 동적 메모리 할당에 의한 지연을 제거한다. 비디오 응용은 이 기법을 사용하여, 디지털 비디오 데이터를 물리적 디스크로부터 직접 버퍼안에 위치시킨 후 상기 프로세스내에서 비디오 데이터를 여러 번 이동시킴없이 비디오 테이프를 직접 출력 장치로 전송함으로써 데이터 이동을 최소화할 수 있다.

[G. 비디오 응용에 최적화된 디스크 드라이브(Disk drive optimized for video applications)]

비디오 스트림들은 등시적으로(isochronously) 즉 인간의 눈으로 보아 움직임의 불연속으로 감지되거나 귀로 들었을 때 소리의 중단(interruption in sound)으로 감지될 수 있는 지연(delay)없이 그 목적지까지 전달되는 것이 중요하다. 현재의 디스크 기술은 데이터 액세스시 심각한 지연을 초래할 수 있는 주기적 동작(periodic action), 예를 들면 예상 고장 분석 수행(predictive failure analysis)과 같은 동작을 수반할 수 있다. 대부분의 I/O 동작이 100 ms내에 완료되기는 하지만, 100 ms의 주기적 지연도 일상적인 것이며 3초간의 지연이 발생할 수 있다.

매체 스트리머(10)는 또한 높은 데이터 전송 속도를 효율적으로 유지할 수 있어야 한다. 일반적인 목적의 데이터 저장 및 검색을 위해 구성된 디스크 드라이브는, 비디오 서버 응용에 최적화되지 않는다면 메모리, 디스크 버퍼, SCSI 버스 및 디스크 용량의 사용에 있어 비효율성을 겪을 것이다.

본 발명의 일면에 따르면, 본 발명에 이용된 디스크 드라이브는 디스크 파라미터들을 최적화함으로써 대량의 데이터를 평탄하게 또한 적시에(smooth and timely) 전달하도록 맞추어진다. 상기 파라미터들은 비디오 서버들을 위해 특화된(specialized) 디스크 드라이브의 제조에 아예 포함될 수도 있고, 명령 메카니즘(command mechanism)을 통해 설정될 수 있는 변수(variable)들일 수도 있다.

주기적인 동작들을 제어하는 파라미터들은 지연을 최소화하거나 제거할 수 있도록 설정된다. 버퍼 사용에 영향을 주는 파라미터들은 단일의 판독 또는 기록 동작으로 매우 큰 양의 데이터를 전송할 수 있도록 설정된다. SCSI bus와 프로세서 서브 사이의 속도 정합(speed matching)에 영향을 주는 파라미터들은, 데이터 전송이 너무 빠르지도 너무 늦지도 않게 시작되도록 조정(tune)된다. 디스크 매체 자체는 유효 용량 및 대역폭을 최대화하는 섹터 크기로 포맷된다.

최적화를 달성하기 위해서는: 물리적 디스크 매체는 가능한 최대의 물리적 섹터 크기로 포맷된다. 이 포맷팅 옵션(formatting option)은 섹터들 사이의 갭(gap)들에 의한 공간의 낭비를 최소화하며, 장치 용량(device capacity)과 버스트 데이터 속도(burst data rate)를 최대화한다. 바람직한 실시예에서는 섹터 크기가 744 비트이다.

디스크는 그와 관련된 버퍼를 구비할 수 있다. 이 버퍼는 데이터 전송을 위해 버스가 이용될 수 있을 때 디스크 매체로부터 비동기적으로 데이터를 판독하는 데 사용된다. 또한 상기 버퍼는 데이터가 디스크 매체에 전달될 때 버스로부터 등시적으로 도착하는 데이터를 보유(hold)하는데 사용된다. 상기 버퍼는 여러 세그먼트들로 분할될 수 있으며 그 세그먼트들의 개수는 하나의 파라미터에 의해 제어된다. 세그먼트들이 너무 많은 경우는, 각 세그먼트가 너무 작아 단일 전송에서 요구되는 데이터 양을 보유할 수 없게 된다. 버퍼가 가득차면, 디바이스는 재접속(reconnection)을 개시하고 전송을 시작한다. 버스 및 장치를 상기 시점에서 이용할 수 없다면 회전 지연(rotational delay)이 뒤따른다. 본 발명의 바람직한 실시예에서 이 값은 어떠한 버퍼 세그먼트라도 최소한 그 데이터 전송 크기만큼은 코도록, 예를 들면 1로 설정된다.

판독에 의해 버퍼 세그먼트가 채워지기 시작함에 따라, 디스크는 호스트로의 데이터 전송을 달성하기 위해 버스에 대한 재접속을 시도한다. 디스크가 상기 재접속을 시도하는 시점은 버스 사용 효율에 영향을 미친다. 버스 및 디스크의 상대적인 속도가, 상기 버퍼를 채우는 동작(fill operation)중에 호스트로의 데이터 전송을 시작할 가장 좋은 시점을 결정한다. 이와 유사하게 기록 동작 중에는 호스트로부터 데이터가 도달함에 따라 버퍼가 채워질 것이며, 이렇게 채워지는 과정중 어떤 시점에서 디스크는 버스에 대한 재접속을 시도해야 한다. 정확한 속도 정합(speed matching)이 되면 SCSI 버스상에 연결 해제/재선택 사이클(disconnect/reselect cycle)을 더 적게 하여 결과적으로 최대의 스루풋(throughput)을 갖게 한다.

재접속이 시도될 시점을 제어하는 파라미터들은 판독 버퍼 풀 비율(read buffer full ratio) 및 기록 버퍼 엠티 비율(write buffer empty ratio)라 불린다. 비디오 데이터를 위해서는, 이들 비율을 계산하는 바람직한 알고리즘(수식)은 $256 \times (\text{즉시적인 SCSI 데이터 전송 속도} - \text{유지할 수 있는 디스크 데이터 전송 속도}) / \text{즉시적 SCSI 데이터 전송 속도}$ 이다. 버퍼-풀(buffer-full) 및 버퍼-엠티(buffer-empty) 비율로 최근에 선호되는 값은 약 204이다.

몇몇 디스크 드라이브 설계는 온도에 따라 변화하는 헤드의 위치를 주기적으로 재조정(recalibration)하는 것을 필요로 한다. 이런 디스크 드라이브 유형들중 몇몇은, 이러한 온도 보상(thermal compensation)이 하나의 집합내의 모든 헤드들에 동시에 행해질 것인지 아니면 한 번에는 하나의 헤드에만 행해질 것인지를 제어할 수도 있게 한다. 만약 동시에 모든 헤드들에 대해 온도 보상이 행해진다면, 비디오 데이터의 한 번의 판독 동작 동안 수백 밀리초(millisecond)의 지연이 수반될 수 있다. 판독 시간이 오래 지연되면 데이터의 흐름을 원활히 하고 멀티미디어 표현의 인위적 요소(artifact)를 예방하기 위해 결과적으로 더 큰 주 메모리 버퍼를 사용해야 한다. 바람직한 방법은 한

번에 하나의 헤드를 온도 보상할 수 있도록 온도 보상 헤드 제어 함수(thermal compensation head control function)를 프로그램화하는 것이다.

에러 로그(error log)의 저장(saving) 및 예상 실패 분석(predictive failure analysis) 작업은 그 종료까지 수 초가 걸릴 수 있다. 이러한 지연을 제거하고 멀티미디어 표현내의 인위적 요소를 예방하기 위한 매우 큰 주 메모리버퍼를 사용하지 않는다면, 상기와 같은 지연은 비디오 서버 응용에서 허용될 수 없다. 유휴 시간 기능의 제한(Limit Idle Time Function) 파라미터들이 에러 로그 저장 및 유휴 시간 기능의 수행을 막는데 사용될 수 있다. 본 발명의 바람직한 실시예에서는 이들 기능을 제한하기 위해 하나의 파라미터를 설정한다.

[H. 비디오 데이터를 위한 데이터 스트라이프화(Data Striping for Video data)]

비디오 응용들에 있어서는, 동일한 데이터(예를 들면 영화)로부터 다수의 스트림들을 전달할 필요성이 존재한다. 이러한 요건은 데이터를 높은 데이터 속도로 판독할 필요성으로 바뀌어 수 있다. 즉, 하나의 스트림을 전달하는데 필요한 데이터 속도보다 동일한 데이터를 동시에 액세스하는 스트림들의 개수를 곱한 데이터 속도를 필요로 한다. 종래에는 다수의 데이터 사본을 구비하여 상기 문제를 해결함으로써 추가적으로 비용이 지출되는 결과를 낳았다. 본 발명의 매체 스트리머(10)는 단일 데이터 사본으로부터 많은 동시 스트림을 지원(serving)하는 기법을 사용한다. 이 기법은 개별 스트림의 데이터 속도와, 데이터를 동시에 액세스하는 스트림의 개수를 모두 고려한다.

전술한 데이터 스트라이프화(data striping)는, 스트라이프(stripe)라 불리는 다수의 파일 구성 요소(file component)에 존재하도록 그 데이터가 분할된 논리적 파일(logical file) 개념과 관련된다. 각 스트라이프는 서로 다른 디스크 볼륨에 존재할 수 있기 때문에, 논리적 파일이 다수의 물리적 디스크(physical disk)에 걸쳐 수 있다. 상기 디스크들은 근거리(근거리)에 있을 수도 있고(local) 원거리에 있을 수도(remote) 있다.

데이터가 논리적 파일에 기록되면, 그 논리적 파일은 스트라이프들에 순차적으로 배치될 논리적 길이(즉 세그먼트)로 분리된다. 제16도에 도시된 바와같이, 하나의 비디오(비디오 1)를 위한 논리적 파일은 지정 크기, 예를 들어 256KB 로 된 M 개 세그먼트 또는 블록들로 분리된다. 가장 마지막 세그먼트는 데이터가 일부분만 채워졌을 수도 있다. 데이터의 한 세그먼트가 처음의 스트라이프에 배치되며 그 다음 세그먼트는 두 번째 스트라이프에 배치되고 그 뒤로 같은 방식으로 배치된다. 각 스트라이프마다 세그먼트가 하나씩 기록되고 나면 그 다음 세그먼트는 맨 처음 스트라이프에 기록된다. 그러므로 하나의 파일이 N 개의 스트라이프로 나누어지면 (striped) 스트라이프 1은 세그먼트 1, N+1, 2*N+1 등을 포함하며 스트라이프 2는 세그먼트 2, N+2, 2*N+2 등을 포함하고 그 다음도 마찬가지이다.

위와 유사한 데이터 스트라이프화 기법이 RAID 배열의 데이터 처리에서도 사용되는 것으로 알려졌으며, 여기서의 스트라이프화하는 목적은 하나의 디스크가 손실되는 경우에도 데이터의 완전성(data integrity)을 보장하기 위한 것이다. 그러한 RAID 저장 장치 시스템은 N 개의 디스크들중 하나를 데이터 복구(data recovery)시에 사용되는 패리티 데이터(parity data) 저장에 전용(dedicate)시킨다. 매체 스트리머(10)의 디스크 저장 노드(16)는 RAID와 유사한 구조(RAID-like structure)로 편성되지만 패리티 데이터는 요구되지 않는다(이는 비디오 데이터의 사본이 테이프 저장 장치로부터도 이용 가능하기 때문이다).

제17도는 상기 데이터 배열의 가장 중요한 면을 도시한다. 즉 이용 가능한 디스크 드라이브들에 걸쳐 흩어져 있는 데이터 블록 또는 세그먼트들로 각 비디오 표현을 분할하여, 다수의 사본에 대한 요구없이도 각 비디오 표현이 다수의 드라이브들로부터 동시에 액세스될 수 있게 하는 면을 도시한다. 그러므로 스트라이프화의 착상(concept)이 유종 하나는 본래 데이터 완결성이나 성능의 이유때문이라 아니라 병존성(concurrency) 또는 대역폭(bandwidth) 때문이다. 그래서, 매체 스트리머(10)는 바이트 블록등의 단위가 아닌 재생 세그먼트(play segment) 단위로 비디오 표현을 스트라이프화 한다.

제17도에 보인 바와같이, 비디오 데이터 파일 1은 M개 세그먼트로 분할되어 4개의 스트라이프들로 나누어져 스트라이프 1은 비디오 파일 1의 세그먼트 1, 5, 9 등을 포함하는 파일이며, 스트라이프 2는 비디오 파일 1의 세그먼트 2, 6, 10 등을 포함하는 파일이며, 스트라이프 3은 비디오 파일 1의 세그먼트 3, 7, 11 등을 포함하는 파일이며, 스트라이프 4는, 비디오 파일 1의 세그먼트 4, 8, 12 등을 포함하는 파일이며 이는 비디오 파일 1의 M개 세그먼트를 모두가 4 개의 스트라이프 파일중 하나에 포함될 때까지 계속된다.

전술한 스트라이프화 전략이 주어진다면, 각 개별 비디오마다 상기 스트라이프화를 개별화(customize)시키기 위해 다음과 같은 방법으로 파라미터들이 계산된다.

먼저, 디스크로부터 합당한 정도의 유효 데이터 속도를 얻도록 세그먼트 크기가 선택된다. 그러나 세그먼트의 크기는 대기 시간에 악영향을 줄 정도로 클 수는 없다. 또한 그 크기는 메모리내에서 버퍼링/캐쉬를 할 수 있을 정도로 충분히 작아야 한다. 바람직한 세그먼트 크기는 256KB이며, 128KB/초 내지 512KB/초 범위내 데이터 속도의 비디오 표현에 대해서는 일정하다. 만약 비디오 데이터 속도가 이보다 더 높다면 더 큰 세그먼트 크기를 사용하는 것이 바람직할 것이다. 세그먼트 크기는 동일한 매체에 저장된 비디오 표현의 범위 동안 I/O 동작의 기본 단위(basic unit)에 의존한다. 본 실시예에서 사용된 원칙은 대략 0.5 내지 2 초간의 비디오 데이터를 포함하는 세그먼트 크기를 사용하는 것이다.

그 다음은 스트라이프들의 개수, 즉 비디오 데이터가 분포될 수 있는 디스크들의 개수가 결정된다. 이 숫자는 요구되는 총 데이터 속도를 유지할 수 있을만큼 충분히 커야하며, 예측되는 사용 속도에 기초하여 각 비디오 표현마다 개별적으로 계산된다. 특히, 각 디스크는 자신과 관련된 논리적 볼륨(logical volume)을 갖는다. 각 비디오 표현은 필요한 스트라이프 숫자만큼의 구성 요소 파일(component file)들로 나뉘어진다. 각 구성 요소 파일은 서로 다른 논리적 볼륨에 저장된다. 예를 들어 비디오 데이터가 스트림당 250KB/초의 속도로 전달되어야 하고 동일한 비디오로부터 30 개의 동시 스트림들이 예를 들어 15 초 간격으로 지원된다면, 전체 데이터 속도는 7.5MB/초가 얻어진다. 만약 디스크 드라이브가 평균적으로 3MB/초를 지원할 수 있다면 그 비디오 표현에 최소한 3개의 스트라이프들이 요구된다.

데이터가 디스크로부터 판독될 수 있는 유효 속도는 판독 동작의 크기에 영향을 받는다. 예를 들어 데이터가 디스크로부터 4KB 블록들로(디스크상의 랜덤한 위치로부터) 판독된다면, 유효 데이터 속도는 1 MB/초인 반면, 데이터가 256KB 블록들로 판독된다면 유효 데이터 속도는 3MB/초일 것이다. 그러나 데이터가 매우 큰 블록들로 판독된다면 버퍼로 요구되는 메모리의 크기도 또한 증가하며, 대기시간 즉 판독된 데이터를 사용함에 있어서의 지연 또한 증가하는데, 이는 그 데이터가 액세스될 수 있기 전에 상기 동작이 종료되어야만 하기 때문이다. 그러므로 데이터 전송을 위한 크기를 선택하는데는 트레이드-오프(trade-off)가 존재한다. 데이터 전송의 크기는 장치들의 특성 및 메모리 구성에 기초하여 선택된다. 바람직하게는 데이터 전송의 크기가 곧 선택된 세그먼트 크기이다. 주어진 세그먼트 크기에 대해, 장치 하나로부터의 유효 데이터 속도가 결정된다. 예를 들어서 몇몇 디스크 드라이브에 대해서는 256KB 의 세그먼트 크기가 디스크 드라이브의 효과적 사용 및 버퍼 크기(256KB)간에 훌륭한 균형을 제공한다.

만약 스트라이프화가 사용되지 않는다면, 지원될 수 있는 최대 스트림의 수는 디스크의 유효 데이터 속도에 의해 제한되어, 예를 들어서 유효 데이터 속도가 3 MB/초이고 스트림 데이터 속도가 200 KB/초라면 단지 15개 스트림만이 디스크로부터 공급될 수 있다. 만약 동일한 비디오에 대해 60개의 스트림이 필요하다면 데이터는 4 개의 디스크들에서 중복되어야(duplicated)한다. 그러나, 본 발명에 따라 스트라이프화가 이용된다면, 상기 용량의 1/4을 가진 4개 디스크가 사용될 수 있다.

비디오 데이터의 단일 사본으로부터의 총 60 개 동시적 스트림들을 위해, 4개 스트라이프들 각각으로부터 15개 스트림들이 동시에 재생될 수 있다. 스트림들의 시작 시간은, 60개 스트림에 대한 요청들이 스트라이프들간에 균등하게 퍼져있도록 하기 위해 스큐(skew)된다. 만약 스트림들이 서로 근접한 시간에 시작된다면, 캐쉬된 비디오 데이터를 사용함으로써 I/O 에 대한 요구를 줄일 수 있음에도 주목하자.

주어진 비디오에 대한 스트라이프들의 개수는 두가지 인자(factor)에 의해 영향을 받는데, 첫 번째 인자는 그 비디오로부터 언제든지 공급될 수 있는 스트림의 최대개수이며 두 번째 인자는 상기 비디오와 동일한 디스크상에 저장된 모든 비디오들로부터 항상 공급되어야 할 스트림의 총 개수이다.

하나의 비디오에 대한 스트라이프(들)의 개수는 다음과 같이 결정된다 : $s = \text{maximum}(r*n/d, r*m/d)$

여기서

r = 스트림이 재생되어야 하는 명목상의 데이터 속도

n = 상기 명목상 데이터 속도로 상기 비디오 표현으로부터 나오는 동시적 스트림들의 최대 개수

d = 디스크로부터의 유효 데이터 속도(디스크로부터의 유효 데이터 속도는 세그먼트 크기에 의해 영향을 받는다는 사실에 주목하자).

m = 상기 비디오 표현중 일부분이라도 포함하고 있는 모든 디스크들로부터 명목상의 데이터 속도로 나오는 동시적 스트림들의 최대 개수

s = 비디오 표현을 위한 스트라이프들의 개수

비디오 표현을 위한 데이터가 스트라이프화되어 있는 디스크들의 개수는 하나의 세트(set)로 관리되어, 매우 큰 하나의 물리적 디스크로 생각할 수 있다. 스트라이프화는, 한 비디오 파일의 크기가 시스템의 물리적 파일 시스템이 허가할 수 있는 가장 큰 파일에 의한 크기 제한을 초과할 수 있게 한다. 비디오 데이터는 일반적으로 세트내의 모든 디스크들에 대해 항상 똑같은 크기의 저장 장소를 요구하지는 않을 것이다. 디스크 사용의 균형을 맞추기 위해서는, 비디오가 스트라이프화될 때, 가장 큰 미사용 공간(free space)을 가진 디스크로부터 스트라이프화가 시작된다.

예를 들어 2M비트/초(250,000 바이트/초)의 속도를 재생되어야 하는 즉 r 이 250,000바이트/초인 비디오 표현의 경우를 고려하되, 이 비디오로부터 30개 까지의 동시적 스트림들이 전달되어야 하는 즉 n 이 30인 경우를 가정하자. 또한 상기 예에서 m 은 30, 즉 모든 디스크로부터 전달되어야 하는 스트림들의 총수도 30 이라고 가정하자. 더 나아가 250,000 바이트 크기의 세그먼트들로 데이터가 분할되며 주어진 세그먼트 크기(250,000 바이트)를 위한 디스크로부터의 유효 데이터 속도는 3,000,000 바이트/초라고 가정하자. 그러면 필요한 스트라이프들의 수 n 은 $2.5(250,000 \times 30 / 3,000,000)$ 이며, 이는 3으로 반올림된다($s = \text{Ceiling}(r \times n / d)$).

상기 비디오 데이터를 포함하는 모든 디스크들로부터의 최대 스트림 개수가 예를 들어 45 라면 $250,000 \times 45 / 3,000,000$ 또는 3.75 개, 반올림하여 4개의 스트라이프가 필요하다.

상기 비디오를 3개 스트라이프로 분할하는 것이 단일 비디오 사본으로부터 30 개 스트림을 전달하는 요건을 만족하기에 충분하지는 않지만, 만약 그 비디오를 포함하는 디스크가 다른 내용도 포함하고 있고, 상기 비디오로부터 지원되어야 할 스트림 총수가 45 라면, 4 개의 디스크 드라이브가 필요하다(스트라이프화 레벨이 4 이다).

매체 스트리머(10)에서 상기 알고리즘이 사용되는 방식은 다음과 같다. 기억 장소(디스크 드라이브의 개수)는 디스크들의 그룹으로 나뉜다. 각 그룹은 주어진 숫자 만큼의 동시적 스트림들을 전달하기 위해(디스크마다 소정의 세그먼트 크기에 근거한 유효 데이터 속도로) 일정한 용량 및 능력을 갖는다. 각 그룹을 위한 세그먼트 크기는 일정하다. 서로 다른 그룹은 서로 다른 세그먼트 크기를 선택할 수 있다(그러므로 서로 다른 유효 데이터 속도를 가진다). 비디오 표현 하나가 스트라이프화되면, 후술하는 판단 기준(criteria)에 의해 그룹 하나가 먼저 선택된다.

세그먼트의 크기는 비디오 데이터의 속도와 모순이 없어야 한다. 즉 스트림 데이터 속도가 250,000 바이트/초라면 세그먼트 크기는 125KB 내지 500KB 범위에 있다. 그 다음 기준은 그룹내에 있는 디스크들 수가 동시적 스트림의 최대 개수를 지원할 수 있을 만큼 충분함, 즉 r 이 스트림 데이터 속도이고 n 이 동시적 스트림의 최대 개수, d 가 그룹내에 있는 디스크의 유효 데이터 속도라 할 때 디스크들의 개수만큼을 지원할 수 있을 정도로 충분함이 보증되어야 한다. 최종적으로, 디스크 그룹내 모든 비디오들로부터 지원될 필요가 있는 동시적 스트림들의 총 합계는 (그 그룹의) 용량(capacity)을 초과하지 않음이 보증되어야 한다. 즉 m 이 그 그룹의 용량이라면, $m - n$ 은 그 그룹내에 이미 저장된 비디오들로부터 동시에 재생될 수 있는 모든 스트림들의 합계보다 크거나 같아야 한다.

상기 계산은 비디오 데이터가 매체 스트리머(10)에 적재될 때 제어 노드(18)에서 행해진다. 가장 간단한 경우는 모든 디스크들이 단일 풀(single pool)에 있는 것으로, 이 단일 풀은 기억 장소 및 지원 가능한 스트림 개수 모두에 대해 매체스트리머(10)가 갖는 총 용량을 정의한다. 이런 경우 주어진 개수의 동시적 스트림을 지원하기 위해 필요한 디스크들(스트라이프들)의 개수는, m 을 스트림 개수라 하고 r 을 스트림 하나의 데이터 속도라 하며 d 를 디스크 하나의 유효 데이터 속도라 할 때 공식 $m \times r / d$ 에 의해 계산된다. 만약 스트림들이 서로 다른 데이터 속도를 가질 수 있다면 상기 공식 중 $m \times r$ 부분이 Max (모든 동시적 스트림들의 데이터 속도에 대한 합계)로 대체되어야 한다.

데이터를 기록하는데 상기 기법을 사용하면, 결과적으로 비디오 표현의 디지털 표시에 대한 다수의 사본이 필요하지 않고도 지정 데이터 속도로 많은 스트림을 전달하기 위해 데이터가 판독될 수 있다. 다수의 디스크 볼륨에 걸쳐 데이터를 스트라이프화함으로써, 한 스트림 전달을 위해 파일의 한 부분을 판독하는 작업은 다른 스트림 전달을 위해 파일의 다른 부분을 대한 판독하는 작업을 방해하지 않는다.

[1. 매체 스트리머의 데이터 전송 및 변환 절차(Media streamer data transfers and conversion procedure)]

[1. 1. 스위치(18)로의 비디오 전달을 위한 동적 대역폭 할당]

통상적으로 비디오 서버는 두 가지 프로파일(profile)들 중 하나에 일반적으로 맞게(fit) 되어 있다. 즉 비디오 서버

는, 저렴한(그러므로 대역폭이 낮은) 비디오 서버를 구축하기 위해 PC 기술을 사용하거나, 높은 대역폭의(그러므로 값도 비싼) 비디오 서버를 구축하기 위해 슈퍼-컴퓨터 기술을 사용한다. 본 발명의 목적은 슈퍼-컴퓨터 기술에 소요되는 높은 비용없이도 높은 대역폭의 비디오를 전달하는데 있다.

제18도는 스위치 인터페이스 및 저장 노드간의 통상적인 논리적 연결(logical connection)을 보인다. 이 스위치 인터페이스는 저장 노드 안팎으로 모두 비디오(및 제어 정보)가 전달될 수 있도록 전 이중 방식(full duplex)(즉, 정보가 양 방향으로 동시에 전달될 수 있다)이어야 한다. 비디오 내용은 저장 노드에 한 번 기록된 후 여러 번 판독되기 때문에, 저장 노드에 대한 대부분의 대역폭 요구는 스위치를 향하는 방향쪽에 있다. 전형적인 스위치 인터페이스의 경우 저장 노드의 대역폭은 제대로 활용되지 못하는데(under-utilized), 이는 기록 능력에 전용되는 절반의 대역폭이 별로 사용되지 않기 때문이다.

제19도는 본 발명에 따른 스위치 인터페이스를 보인다. 이 인터페이스는 노드(여기서는 본보기로 저장 노드(16)가 사용되었다)에 대한 현재의 수요를 충족시키기 위해, 스위치(18) 안으로 또는 스위치에서 밖으로 총 대역폭을 실시간으로 동적 할당한다(dynamically allocate). 통신 노드(14)도 위와 유사한 요건을 갖지만 대부분의 대역폭이 스위치(18)로부터 나오는 방향에 존재한다.

전술한 동적 할당은, 스위치(12)를 위한 적절한 라우팅 헤더(routing header)를 사용하여 두 개 또는 그 이상의 물리적 스위치 인터페이스들을 하나의 논리적 스위치 인터페이스(18a)로 그룹화(grouping)함으로써 달성된다. 그러면 비디오 데이터는(예를 들어 판독시) 두 개의 물리적 인터페이스들 사이에서 분할(split)된다. 전술한 바와같이 다수의 저장 장치 유닛간에 데이터를 스트라이프화함으로써 상기 분할이 용이해진다. 수신 노드(receiving node)는 비디오 데이터를 다시 하나의 논리적 스트림으로 결합시킨다.

예를 들어, 제18도에서의 스위치 인터페이스는 2X MB/초의 전이중 방식 즉 각 방향이 X MB/초이다. 그러나 비디오 데이터는 보통 한 방향으로만(저장 노드에서 스위치쪽으로) 전송된다. 그러므로 비록 상기 노드가 그 보다 두배의 능력(2X)을 갖고 있다 하더라도, 저장 노드로부터 단지 X MB/초의 비디오 대역폭만 전달될 뿐이다. 저장 노드는 제대로 활용되지 못하고 있는 것이다. 제19도의 스위치 인터페이스는 비디오를 저장 노드에서 스위치로 전송하는데 있어 2X MB/초 대역폭 전체를 동적으로 할당한다. 그 결과, 노드로부터의 대역폭이 증가하며 비디오 서버로부터의 대역폭이 높아지고 비디오 스트림당 소요되는 비용은 낮아진다.

[J. 통신 어댑터를 사용한 등시적 비디오 데이터 전송(Isochronous video data delivery using communications adapters)]

디지털 비디오 데이터는 순차적이고, 연속적이며, 대량이며, 내용중심적(content critical)이라기 보다는 시간중심적(time critical)이다. 비디오 데이터의 스트림은 높은 비트 속도에서 동시적으로 전달되어야 하므로, 필수적이 아닌 모든 오버헤드는 데이터 경로(data path)에서 최소화될 것이 요구된다. 대개, 수신용 하드웨어는 비디오 세트 탑 박스(video set top box)이거나 다른 적당한 비디오 데이터 수신기(video data receiver)이다. 표준 직렬 통신 프로토콜(standard serial communication protocol)에서는 동기화 및 데이터 검증을 위해 종종 하드웨어 레벨에서 스트림내에 추가적으로 데이터를 몇 비트 또는 몇 바이트씩 삽입한다. 만약 수신기가 상기 추가적인 데이터를 투명하게 (transparently) 제거하지 못한다면, 전술한 데이터 추가는 비디오 데이터 스트림을 망쳐놓게 된다. 상기 비트들 또는 바이트들에 의해 도입되는 추가적인 오버헤드도 또한 유효 데이터 속도를 감소시켜, 비디오의 압축해제 및 변환에러(decompression and conversion error)가 생기게 한다.

표준 통신 어댑터를 통한 비디오 데이터 전송은 사용자에게 등시적으로 데이터를 전달하는 것을 보장하려면 표준 직렬 통신 프로토콜의 속성 중 대부분을 쓸 수 없게(disabling)함을 요구한다고 알려져 왔다. 이를 달성하기 위한 방법은 사용되는 통신 어댑터에 따라서 다르지만, 기본이 되는 개념은 다음과 같다. 제20도에서, 통신 노드(14)에 있는 직렬 통신 칩(200)은 패리티 비트나 시작 및 정지 비트, 순환 여유 검사 코드(cyclic redundancy check code) 및 동기 바이트(sync bytes)와 같은 데이터 포매팅 정보 및 데이터 완전성을 위한 정보들을 쓸 수 없게 하며 유힬 문자(idle character)들의 생성을 막는다. 데이터 블록을 적재하기 위한 버스 사이클을 허가하는 동안에도 일정한(등시적인) 출력 비디오 데이터 스트림을 보장하기 위해서, 입력 FIFO 버퍼들(202,204,206등)이 사용된다. 1000 바이트의 FIFO 버퍼(208) 하나가 CPU 및 버스의 적재 논리를 간단하게 할 수 있다.

만약 통신 출력 칩(200)이 초기의 동기 바이트(sync byte) 생성의 억제(disabling)를 허가하지 않는다면, 상기 동기 바이트의 값은 각 데이터 블록의 첫 번째 바이트값으로 프로그램화 된다(그리고 데이터 블록 포인터는 두 번째 바이트로 증가된다). 바이트 정렬(byte alignment)은 실제 데이터(real data)에 행해져야만 하는데, 그 이유는 임의로

첨가된 바이트(padding byte)들은 이들이 실제로 압축된 비디오 데이터가 아닌 경우 데이터 스트림을 망칠 것이기 때문이다.

압축된 비디오 데이터의 높은 품질 수준을 위해 요구되는, 일정하고 고속인 직렬 데이터 출력을 달성하기 위해서는, 하나의 순환 버퍼(circular buffer) 또는 다수의 대형 버퍼들(예를 들면 202,204,206)이 사용되어야만 한다. 이는 그 전에 채워진 버퍼들로부터 데이터를 출력하는 동안 입력 버퍼를 채우기 위한 충분한 시간을 허가하기 위해서 필요하다. 비디오 데이터 스트림 경로의 초기에 버퍼 패킹(buffer packing)이 행해지지 않는다면, 비디오 종료 조건(end of video condition)은 다음 버퍼 전송이 종료되기 전에 출력될 매우 작은 버퍼에서 데이터의 언더런(data underrun)을 초래할 수 있다. 이 사실에 의해 최소한 3개의 독립적인 대형 버퍼들이 요구된다. 2중 모드 메모리(dual mode memory)(판독하는 중에도 기록할 수 있는)내에 있는 순환 버퍼를 사용하는 것도 적절한 구현이 될 수 있다.

[J1. 압축된 비디오 영상 또는 영화를 MPEG-1, 1+, 또는 MPEG-2 디지털 데이터 포맷으로부터 산업계 표준 텔레비전 포맷(NTSC 또는 PAL)으로 변환]

전술한 바와같이, 디지털 비디오 데이터는 디스크로부터 버퍼 메모리로 이동된다. 일단 버퍼 메모리에 충분한 양의 데이터가 있기만 하면, 그 데이터는 메모리로부터 통신 노드(14)내의 인터페이스 어댑터로 이동된다. 사용되는 인터페이스는 SCSI 20 MB/초 고속/광역(fast/wide) 인터페이스이거나 SSA 직렬 SCSI 인터페이스이다. SCSI 인터페이스는 15개 주소를 처리할 수 있게 확장되며 SSA 구조는 256 개까지 주소를 지원할 수 있다. 다른 적절한 인터페이스로는 RS 422, V.35, V.36 등을 들 수 있다. 다른 적절한 인터페이스로는 RS 422, V.35, V.36 등을 들 수 있다.

제21도에 보인 바와같이, 인터페이스로부터의 비디오 데이터는 통신 버스(210)를 통하여 통신 노드(14)로부터 데이터가 버퍼링될 NTSC 어댑터(212)(제20도 참조)로 전달된다. 어댑터(212)는 로컬 버퍼(local buffer)(214)로부터 데이터를 가져오는데, 이 로컬 버퍼에는 버스의 성능을 최대화하기 위해 다수의 데이터 블록들이 저장되어 있다. 어댑터(212)의 주요 목적은 메모리(214)에서 MPEG 칩들(216,218)까지 또 NTSC 칩(220) 및 D/A(222)까지 동시적 데이터 흐름을 유지하여, 비디오 및/또는 오디오 전달에 있어 중단(interruption)이 없음을 보증하는 것이다.

MPEG 논리 모듈(216,218)은 디지털(압축된) 비디오 데이터를 구성 요소 레벨(component level)의 비디오 및 오디오로 변환한다. NTSC 인코더(220)는 상기 신호를 NTSC 베이스밴드 아날로그 신호로 변환한다. MPEG 오디오 디코더(216)는 디지털 오디오를 병렬 디지털 데이터로 변환하며 이 병렬 디지털 데이터는 디지털 대 아날로그 변환기(222)를 통과하고 좌측 및 우측 오디오 출력(audio Left and Right outputs)을 생성하도록 필터링된다.

속도 정합 및 동시적 전달 문제를 해결책을 만드는데 있어서의 목표는 시스템의 대역폭 전달을 최대화할 뿐 아니라 최소의 성능 제약조건(performance constraints)을 부과하는 접근법이다.

통상적으로 응용 개발자들은 디스크 파일, 테이프 파일, 광학 저장장치 유닛(optical storage unit) 등과 같은 기계적 저장 장치와 프로세서 사이에 데이터를 전달하고 제어하기 위해 SSA 및 SCSI 와 같은 버스 구조를 사용해 왔다. 이 버스들은 모두, 비디오 데이터의 속도 및 동시적 전달을 제어하기 위해 사용되는 수단이 있기만 한다면 비디오 데이터의 높은 대역폭 전달에 적합하게 되는 속성을 가지고 있다.

SCSI 버스는 20M바이트/초 속도로 데이터가 버스트화(bursting)되는 것을 허가하는데, 이는 임의의 비디오 신호 하나가 버퍼 메모리로부터 지정 NTSC 어댑터로 이동되는 시간을 최소화시킨다.

어댑터 카드(212)는 하나의 대형 버퍼(214)를 포함하는데, 이 버퍼는 최고의 속도로 데이터를 버스(210)에서 메모리로 버스트화시킬 수 있으며 NTSC 디코더 칩(216,218)으로의 전달을 위해 훨씬 낮은 속도로 데이터를 버퍼(214)로부터 제거하는 수행 능력(performance capability)을 가진다. 버퍼(214)는 또한 더 작은 버퍼들로 나뉘어 지고(segmented), 이들은 순환적인 방식으로 연결된 다수의 버퍼들로 동작하기 위해 소프트웨어 제어부(software controls)를 통해 연결된다.

이는 시스템으로 하여금 다양한 블록 크기를 가진 데이터들을 개별적인 버퍼들에 전달하며 재생 순서를 제어한다. 이러한 접근법의 장점은 비디오 데이터에 대한 임의의 요구가 있기 전에 고속으로 미리 비디오 데이터 블록들을 전달하는 임무로부터 시스템 소프트웨어를 자유롭게 한다는 것이다. 이는 매체 스트리머(10)에게 동적인 스루풋

요건상에서 (dynamic throughput requirement) 다수의 다중 비디오 스트림을 처리할 수 있는 능력을 부여한다. 통신 노드내의 프로세서가 (여유) 시간을 갖는다면, 프로세서는 차례로 재생될 여러 개의 대형 데이터 블록들이 전달되게 할 수 있다. 일단 이 작업이 끝나면, 프로세서는 각 포트로 저속의 연속된 등시적 데이터를 전달할 즉시적 필요성 없이 다른 스트림들을 자유로이 제어할 수 있다.

상기 디코더 시스템의 비용 효율을 개선시키기 위해, 상기 대형 디코더 버퍼(214) 및 MPEG 디코더들(216,218) 사이에 작은 FIFO 메모리(224)를 넣는다. 이 FIFO 메모리(224)는 제어기(226)로 하여금 더 작은 크기의 통상 512 바이트의 데이터 블록들을 버퍼(214)에서 FIFO메모리(224)로 이동시킬 수 있게 하며, 이 FIFO 메모리(224)는 MPEG 디코더(216,218)로의 전달을 위해 데이터를 직렬 비트 스트림들로 변환시킨다. 상기 오디오 및 비디오 디코더 칩들(216,218)은 모두 동일한 직렬 데이터 스트림으로부터 입력을 받아, 필요한 데이터를 내부적으로 분리하고 디코드할 수 있다. FIFO 메모리(224)의 출력으로부터의 데이터 전송은 등시적인 방식, 또는 실질적으로 등시적인 방식으로 행해져서, 비디오 표현의 사용자 또는 소비자에게 중단없는 비디오 표현을 전달함을 보증한다.

[K. SCSI 장치들로의 디지털 비디오 전송]

제22도에 보인 바와같이, 버퍼 메모리로부터 나오는 압축 디지털 비디오 데이터 및 명령 스트림들은 장치 레벨의 소프트웨어(device level software)에 의하여 SCSI 명령들 및 데이터 스트림들로 변환되며, SCSI 버스(210)를 통하여 타겟 어댑터(target adapter)(212)를 향해 SCSI II 고속 데이터 속도로 전송된다. 상기 데이터는 그 후 아날로그 비디오 및 오디오 데이터로 압축 해제 및 변환하기 위해 요구되는 내용 출력 속도로 MPEG 논리부에 버퍼링되고 인가된다(buffered and fed). 데이터 흐름에 보조를 맞추고(pace) 적절한 버퍼 관리를 보증하기 위해, SCSI 버스(210)에 걸쳐 피드백(feedback)이 제공된다.

SCSI NTSC/PAL 어댑터(212)는 SCSI버스(210)에 하이레벨 인터페이스(high level interface)를 제공하여, 표준 SCSI 프로토콜의 일부분(subset)을 지원한다. 정상적인 동작 방식(normal mode of operation)은 어댑터(212)를 개방(open)하고, 데이터 (비디오 및 오디오) 스트림들을 어댑터에 기록하고, 완전히 종료된 경우에만 어댑터(212)를 폐쇄(close)하는 것이다.

어댑터(212)는 그 버퍼를 가득찬 상태로 유지하는데 필요할 만큼의 빠른 속도로 데이터를 가져오며(pull), 통신 노드(14) 및 저장 노드(16)는 버스 데이터 전송을 최적화하고 버스 오버헤드를 최소화할 수 있도록 크기가 정해진 데이터 블록들을 제공한다.

시스템 파라미터들은 필요하다면 모드 선택 SCSI 명령 (Mode Select SCSI Command)을 사용하여 제어 패킷(control packet)을 통해 겹쳐써질(overwritten) 수 있다. 비디오/오디오 동기화는 상기 어댑터(212)의 내부적인 것이며 외부적인 제어는 필요하지 않다. 자동적 재동기화(automatic resynchronization) 및 연속되는 오디오/비디오 출력에 의해 에러가 최소화된다.

[K1. SCSI 레벨의 명령에 대한 설명]

SCSI 비디오 출력 어댑터의 기능성(functionality)에 맞추기 위해, 표준적인 공통 명령들뿐 아니라 직접 액세스 장치 명령과 순차적 장치 명령이 혼합된 것도 사용된다. 모든 SCSI 명령들과 마찬가지로, 매 명령 후에 유효 상태 바이트(valid status byte)가 반환되며, 검사 조건(check condition)이 반환된다면 데이터 감지 영역(sense data area)에 여러 조건이 적재된다. 사용되는 표준 SCSI 명령으로는 RESET, INQUIRY, REQUEST SENSE, MODE SELECT, MODE SENSE, READ, WRITE, RESERVE, RELEASE, TEST UNIT READY가 포함된다.

비디오 명령들(video commands) : 비디오 제어 명령은 사용자 레벨의(user-level) 비디오 출력 제어 명령이며, 앞에서 나열한 표준 명령에서 확장된 것이다. 이 명령들은, 로우 레벨 운영체제 또는 SCSI비디오 출력 어댑터(212)의 인터페이스를 직접 상대하는 SCSI 명령들에게, 더욱 간단화된 사용자 레벨의 프론트 엔드(front end)를 제공한다. 상기 각 명령은 그 구현시에, 필요한 비디오 장치 기능을 에뮬레이트(emulate)하고 유효하지 않은 제어 상태에 의해서 야기되는 비디오 및 오디오의 변칙(anomaly)을 회피하기 위해 마이크로코드(microcode)를 사용한다. 단일의 SCSI 명령인 SCSI 시작/정지 유닛 명령 (SCSI START/STOP UNIT)은, 상기 명령과 함께 이동되는 임의의 필요 파라미터들을 사용해서 비디오 제어 명령을 타겟 SCSI 비디오 출력 어댑터(212)에 맞게 번역(translate)하는데 사용된다. 이는 사용자 응용 인터페이스 및 어댑터 카드(212)의 마이크로코드 모두를 간단하게 한다.

다음과 같은 명령들이 사용된다.

정지(stop)(SCSI 시작/정지 1-파라미터=모드)

(이 명령이 실행되면) MPEG 칩 세트(216,218)로의 데이터 입력이 멈춰지며 (halted), 소리가 안나고(mute)(이하 뮤트라 하자), 비디오는 공백이 된다(blanked)(이하 블랭크라 하자). 파라미터 필드는 정지 모드(stop mode)를 선택한다. 정상 모드(제1모드)는 PLAY 명령이 비디오 스트림내 같은 위치에서 시작될 수 있도록 버퍼 및 위치 포인터가 현재 상태로 유지되는 것이다(remain current). 제2모드(영화 종료 또는 중지 모드)는 버퍼 포인터를 다음 버퍼의 시작 위치로 세팅시키고 현재의 버퍼를 해제(release) 하는 것이다. 제3모드는 제2모드와 마찬가지로 영화 종료 조건을 위한 것이지만 상기 정지(뮤트 및 블랭크)는 데이터 버퍼가 비워질 때까지 지연된다. 제4모드는 일정한 MPEE 디코더 구현과 함께 사용되어, 데이터를 다 소모하고 나면 오디오에는 지연된 정지를 제공하지만 프레임은 가장 유효 프레임으로 고정(freeze)시킨다. 이들 각 경우에 있어서 비디오 어댑터(212)의 마이크로코드는, 비디오 및 오디오 출력이 적당한 경계상에서 중단되어 다음에 (그 경계를 기준으로) 명확히 재개할 수 있도록 하는 정지 지점(stopping point)을 결정한다.

잠시 멈춤(pause)(SCSI 시작/정지 2-파라미터는 없음)

MPEG 칩 세트(216,218)로의 데이터 입력은 정지되며 오디오는 뮤트 상태가 되지만 비디오는 블랭크 상태가 되지 않는다. 이는 MPEG 비디오 칩 세트(216,218)로 하여금 마지막 유효 프레임으로 고정 프레임 (정지 화면)(freeze frame of the last good frame)을 유지하게 한다. 이것은 비디오 튜브(tube)의 번인(burn-in)을 피하도록 제한된다. 정지 명령은 바람직하게는 제어 노드(18)에 의해 발행되지만, 5분내에 아무런 명령도 받지 못한다면 비디오 출력은 자동적으로 블랭크가 된다. 어댑터(212)의 마이크로코드는 버퍼 위치 및 디코더의 상태를 유지하여 나중에 재생(play)으로 순조롭게 전이될 수 있게 한다.

블랭크-뮤트(blank-mute)(SCSI 시작/정지 3-파라미터=모드)

이 명령은 오디오 출력에 영향을 주지않고 비디오 출력을 블랭크로 할 수 있으며 비디오 출력에 영향을 주지않고 오디오 출력을 뮤트로 할 수 있고, 블랭크도 만들고 뮤트시킬 수도 있다. 뮤트와 블랭크 상태 둘 다 모드 파라미터를 사용하여 단일 명령으로 턴 오프(turn off)될 수 있는데, 이는 보다 순조로운 전이와 명령 오버헤드 감소를 가능하게 한다. 이들은 압축 해제 및 아날로그로의 변환 후, 명확하고 순조로운 전이를 보증하기 위한 하드웨어 제어부에 의해 상기 비디오 어댑터(212)상에 구현된다.

저속 재생(slow play)(SCSI 시작/정지 4-파라미터=속도)

이 명령은 MPEG 칩 세트(216,218)로의 데이터 입력 속도를 낮추어 간헐적으로 (intermittently) 프레임을 고정시킴으로써 VCR의 저속 재생 기능을 시뮬레이트한다. 디지털 에러 잡음을 막기 위해 오디오는 뮤트 상태가 된다. 파라미터 필드는 0에서 100까지의 상대적인 속도를 지정한다. 대안적인 구현에 있어서는, 디코드 칩 세트(216,218)의 에러 처리(error handling)를 디스에이블시킨 후, 디코더 칩 세트로의 데이터 클러킹 속도(data clocking speed)를 원하는 재생 속도로 변경한다. 이는 비디오 어댑터의 클럭 구조가 갖는 융통성(flexibility)에 의존한다.

재생(play)(SCSI 시작/정지 5-파라미터=버퍼)

이 명령은 MPEG 칩 세트(216,218)로의 데이터 공급 프로세스를 시작하여, 오디오 및 비디오 출력을 인에이블시킨다. 어떤 버퍼부터 재생 시퀀스를 시작할 것인가를 결정하기 위해 버퍼 선택 번호(buffer selection number)가 전달되며, (버퍼선택 번호가) 영(zero) 값이면 현재 재생 버퍼가 사용되어야 함을 표시한다(이는 통상적인 동작이다). 영이 아닌 값은 어댑터(212)가 정지 모드(STOPPED mode)에 있는 경우에만 허용되며, 만약 잠시멈춤 모드(PAUSED mode)라면 버퍼 선택 파라미터는 무시되며 현재의 버퍼 선택 및 위치를 사용하여 재생이 개시된다.

'재생(playing)' 할 때에, 제어기(226)는 MPEG 칩 세트(216,218)로의 안정된 데이터 스트림을 유지하며 상기 버퍼들을 순차적으로 교대(rotate) 시킨다. 데이터가 적당한 속도로 주소 0에서 시작하여 N 바이트까지 버퍼로부터 MPEG 버스로 판독된 후 제어기(226)는 다음 버퍼로 스위칭하고 데이터 판독을 계속한다. 상기 어댑터 버스와 마이크로코드는, 어댑터 버퍼(214)로의 SCSI 고속 데이터 전송 및 MPEG 압축 해제 칩(216,218)들에 공급되는 출력 FIFO(224)로의 지속적인 적재를 위해서 충분한 대역폭을 제공한다.

고속 전진(Fast Forward)(SCSI 시작/정지 6-파라미터=속도)

이 명령은 VCR의 고속 전진을 에뮬레이트하는 방식으로 데이터를 스캔(scan)하는데 사용된다. 속도 파라미터에 의해 결정되는 두 가지 모드의 동작들이 존재한다. 0 속도는 비디오 및 오디오가 각각 블랭크되고 뮤트되고 버퍼는 플러쉬(flush)되어야 할 (비워져야 할) 고속 전진임을 의미하며, 비디오 스트림내에서 전진된 새로운 위치로부터

데이터가 수신되면 묵시적인 재생(implicit play)이 실행된다. 1에서 10 사이의 정수값은 입력 스트림이 전진되고 있는 속도를 표시한다. 지정된 데이터 속도 평균을 알기 위해, 데이터 블록들 사이를 건너뛴(skipping)으로써 비디오가 '표본화(sampled)'가 된다. 어댑터(212)는 고속 전진 동작을 에뮬레이트하기 위해 대략 정상 속도로 데이터의 일부분을 재생하고 앞으로 점프(jump)한 후 그 다음 부분은 재생한다.

되감기(Rewind)(SCSI 시작/정지 7-파라미터=속도)

이 명령은 VCR의 되감기를 에뮬레이트하는 방식으로 데이터를 역방향으로 스캔하는데(scan backwards) 사용된다. 속도 파라미터에 따라 두 가지 모드의 동작이 존재한다. 0 속도는, 비디오 및 오디오가 각각 블랭크 및 뮤트 상태가 되고 버퍼들이 플러쉬(flush)되어야 하는 빠른 되감기를 의미하며, 비디오 스트림내에서 앞쪽의 새로운 위치로부터 데이터가 수신되면 묵시적인 재생이 실행된다. 1 내지 10의 정수값은 입력 스트림이 되감기고 있는 속도를 표시한다. 비디오는 지정된 평균 데이터 속도를 얻기 위해 데이터 블록들 사이를 건너뛴으로써(skipping) '표본화'된다. 되감기 데이터 스트림은 비디오 스트림내의 매우 초기 위치부터 '표본화된' 작은 데이터 블록들을 집합시킴으로써 만들어진다. 어댑터 카드(212)는 정상 속도로 재생하기 위한 전이 및 동기화를 원활하게 처리하며, 되감기 스캐닝을 에뮬레이트하기 위해 전에 표본화된 부분(next sampled portion)으로 건너뛴다.

[K2. 버퍼 관리(buffer management)]

디지털 비디오 서버는 많은 병행 출력 장치들에게 데이터를 제공하지만, 디지털 비디오 데이터 압축 해제 및 변환은 일정한 데이터 스트림(constant data stream)을 필요로 한다. SCSI 데이터 버스트 모드 전송(data burst mode transmission)을 이용하기 위해 데이터 버퍼링 기법이 사용되어, 데이터의 언더런(underrun)이나 버퍼의 오버런(overrun)을 피하면서도 매체 스트리머(10)로 하여금 최소의 개입(intervention)으로 많은 스트림들을 통해 데이터를 전송되게 한다. SCSI 비디오 어댑터 카드(212)(제21도, 제22도)는 비디오 데이터를 위한 대형 버퍼(214)를 포함하여, 상기 SCSI 버스트 모드 데이터 전송 프로세스를 충분히 이용할 수 있게 한다. 본보기적인 구성(configuration)으로, 순환 버퍼(wrap-around circular buffer)와 같은 국소적 논리부(local logic)에 의해 처리되는 768K 크기의 버퍼를 들 수 있다. 순환 버퍼는 변화하는 데이터 블록 크기들을 동적으로 처리하는 데에 있어서 고정된 길이의 버퍼들보다도 선호되는데, 고정된 길이의 버퍼는 디지털 비디오 데이터를 전송하는 경우 저장 및 관리 오버헤드 모두에 있어서 비효율적이다.

비디오 어댑터 카드(212)의 마이크로코드는 여러 개의 버퍼 포인터(buffer pointer)들을 지원하여 이들로 하여금 데이터의 첫부분(top)과 데이터의 현재 길이뿐 아니라 데이터의 끝부분(last top)을 보유하게 한다. 이는 실패한 전송에 겹쳐쓰도록 재시도(retry)할 수 있게 하거나, 필요하다면 포인터로 하여금 경우 현재의 버퍼내에서 한 바이트 위치에 위치 지정될 수 있게 한다. 데이터 블록의 길이는 전송된 대로 정확하게 유지되어(예를 들면 중간에 있는 논리에 의해 긴 워드 정렬(long word alignment)이 사용되었다더라도 바이트 또는 워드로 지정된다), 디코드 칩 세트(216,218)로의 유효한 데이터 전달을 보증한다. 이 방법은 데이터 버퍼들의 유연한 제어를 허락하면서도 상태 불변 동작(steady state operation)에 의한 오버헤드를 최소화한다.

[K2.1. 버퍼 선택 및 위치 지정(buffer selection and position)]

버퍼들의 세트가 여러 개 요구된다고 가정하면, 동작에 관련된 모든 버퍼들을 위해 다수의 포인터들이 이용가능하다.

예를 들면 한 세트는 재생(PLAY) 버퍼 및 그 버퍼내의 현재 위치를 선택하는데 사용되며, 두 번째 버퍼 세트는 데이터 사전 적재 동작을 위해 기록 버퍼 및 그 버퍼내의 위치(보통 0 이다)를 선택하는데 사용된다. 가변 길이의 데이터 블록들도 지원되기 때문에, 수신된 각 데이터 블록마다 현재의 길이값 및 최대 길이값이 보유된다.

[K2.2. 자동 모드(Automatic mode)]

버퍼 동작은 비디오 어댑터의 제어기(226)에 의해 관리되며, 버퍼의 주소 0에서 시작하여 이용가능한 그 다음 버퍼 공간으로 N 바이트의 데이터를 배치한다. 제어기(226)는 각 버퍼내의 데이터 길이와 그 데이터가 재생됐는지 여부를 추적한다. 충분한 버퍼 공간이 미사용(free)으로 있을 때마다, 상기 비디오 어댑터 카드는 다음의 기록(WRITE) 명령을 받아들이고 데이터를 DMA 방식으로 버퍼에 가져온다. 만약 데이터 블록 전부를 받아들일만큼의 충분한 미사용 공간이 남아 있지 않다면(대개 저속 재생(slow play) 또는 잠시 멈춤(pause) 상태), 상기 기록(WRITE) 명령은 받아들여지지 않으며 버퍼가 가득차다는 복귀 코드(buffer full return code)가 반환된다.

[K2.3. 수동 모드(manual mode)]

각 버퍼 액세스 명령(기록, 삭제등)을 위해 '현재'의 기록 버퍼 및 그 버퍼내에서 위치(보통 0)를 선택하기 위해 위치 지정(LOCATE) 명령이 사용된다. 상기 버퍼 위치는 성공적으로 전송된 가장 마지막 블록의 데이터 첫부분에 대해 상대적이다. 이는 바람직하게는 비디오 스트림의 전이를 처리하는 동안 행해지며, 시스템내의 명령 오버헤드를 최소화하기 위해서 가능한 빨리 자동 모드가 복귀(reactivate)된다.

[K2.4. 에러 관리(Error management)]

디지털 비디오 데이터 전송은, 데이터 처리 응용에 보통 SCSI가 이용되는 랜덤 데이터 액세스 사용과는 다른 에러 관리 요건들을 가진다. 약간의 데이터 손실은 전송의 중단(transmission interruption) 보다는 덜 치명적이므로, 통상의 재시도 및 데이터 유효화 스킴은 수정되거나 디스에이블된다. 통상의 SCSI 에러 처리 프로시저 다음에는 각 명령의 종료시의 상태 (표시) 단계(status phase) 동안 상태 바이트(status byte)가 복귀된다. 상기 상태 바이트는 GOOD(00) 상태를, 타겟 SCSI 칩(227)이 명령을 받을 수 없을 때는 BUSY(8h) 상태를 표시하고, 에러가 발생한 경우에는 CHECK CONDITION(02h) 상태를 표시한다.

[K2.5. 에러 복구(Error recovery)]

SCSI 비디오 어댑터(212)에 제어기(226)는, 에러 및 상태 정보를 적재하기 위하여 CHECK CONDITION 응답에 대해 감지 요청(request Sense) 명령을 발생하고, 복구 절차가 가능한지를 결정한다. 통상의 에러 복구 절차는 에러 상태를 클리어(clear) 하고, 손상된 데이터를 모두 버리고, 가능한 빨리 정상적인 재생을 재개(resume)하는 것이다. 최악의 경우, 어댑터(212)가 리셋(reset)되어야 하며 재생이 재개되기 전에 데이터를 다시 적재해야 할 수도 있다. 에러 상태(error condition)가 기록되며(logged), 이는 다음번의 INQUIRY 또는 REQUEST SENSE SCSI 동작에 의해 호스트 시스템으로 보고된다.

[K2.6. 자동 재시도(Automatic retries)]

버퍼가 가득차거나(buffer full) 장치가 사용중(device busy)인 상황에서는 재시도가 X 번까지 자동적으로 되며, 여기서 X 는 스트림 데이터 속도에 의존한다. 이런 자동 재시도는 다음 데이터 버퍼가 도착하는 시점까지만 허용된다. 상기 시점에서, 상황이 예측할 수 없는 것(즉, 버퍼가 가득찬 상황, 그러나 PAUSED 또는 SLOW PLAY 모드는 아닌)이라면 에러가 기록 되고, 비디오 재생을 복귀하고 계속하기 위해서는 장치의 리셋 (reset) 또는 클리어 (clear) 가 필요할 수도 있다.

비록 본 발명이 사용자에게 비디오 표현을 전달하는 것을 중심으로 설명되기는 하였지만, 양 방향성 비디오 어댑터(bidirectional video adapter)는 비디오 표현을 수신하는데 사용될 수도 있고, 비디오 표현을 그 데이터 표시로서 디지털화(digitize)하는데 사용될 수도 있고, 제어 노드(18)에 의해 지정되는 대로 저 대기시간 스위치(18)를 통해서 저장 노드(들)(16,17)내에 저장하기 위해서 데이터 표현을 통신 노드(14)로 버스(210)를 통해 전송하는데 사용될 수도 있음을 알아야 한다.

전술한 설명은 단지 본 발명의 본보기일 뿐이라는 점이 이해되어야 한다. 본 발명의 범위를 벗어나지 않고도, 본 발명의 기술분야에 있는 당업자들은 여러 가지 변경 및 수정을 할 수 있다. 따라서, 본 발명은 첨부된 특허청구범위의 범위내에 속하는 그러한 모든 변형, 수정들을 포함하는 것으로 의도되었다.

*(57)청구의 범위***청구항1**

다수인 N개의 데이터 블록들로서 저장되는 비디오 표현(video presentation)으로서 그 (비디오 표현의) 전체를 표현하는데 시간 T가 요구되고 상기 각 데이터 블록은 상기 비디오 표현의 대략 T/N 기간에 대응하는 데이터를 저장하는 최소한 하나의 비디오 표현에 대한 디지털 표시(digital representation)를 저장하기 위하여 다수의 대용량 저장 장치 유닛으로 구성된 대용량 저장 장치(mass storage)를 포함하는 최소한 하나의 저장 노드(storage node)와, 상기 최소한 하나의 저장 노드의 출력으로부터 비디오 표현의 디지털 표시를 수신하기 위해 상기 최소한 하나의 저장 노드의 출력에 연결(couple)된 최소한 하나의 입력 포트(input port)와, 상기 디지털 표시의 소비자에게 하나의 데이터 스트림으로 디지털 표시를 각각 전송하는 다수의 출력 포트(output port)들을 각자 구비하는 다수의

통신 노드(communication node)로 구성되며, 상기 디지털 표시의 N개 데이터 블록들은 X개의 스트라이프로 분할되어(partitioned into x stripes) 데이터 블록 1, X+1, 2*X+1, ... 등은 상기 X개 스트라이프들 중 첫 번째 스트라이프와 연관되고 데이터 블록 2, X+2, 2*X+2, ... 등은 상기 X개 스트라이프들 중 두 번째 스트라이프와 연관되며, 상기 X개 스트라이프들 각각은 상기 다수의 저장 장치 유닛들중 서로 다른 저장 장치 유닛에 저장되는 것을 특징으로 하는 매체 스트리머(media streamer).

청구항2

제1항에 있어서, 상기 다수의 대용량 저장 장치 유닛은 비디오 표현의 디지털 표시에 대해 하나의 사본(single copy)을 저장하며, 상기 X개 스트라이프들은 다수의 데이터 스트림들이 상기 N개 데이터 블록들 중 동일한(same) 하나의 블록을 동시에 전달할 수 있게 하는 방식으로 판독되는 것을 특징으로 하는 매체 스트리머.

청구항3

제1항에 있어서, 상기 다수의 대용량 저장 장치 유닛은 비디오 표현의 디지털 표시에 대해 하나의 사본을 저장하며, 상기 X개 스트라이프들은 다수의 데이터 스트림들이 상기 N개 데이터 블록들 중 서로 다른(different) 하나의 블록을 동시에 전달할 수 있게 하는 방식으로 판독되는 것을 특징으로 하는 매체 스트리머.

청구항4

제1항에 있어서, 상기 T/N의 기간은 대략 0.2초 내지 대략 2초의 범위안에 있는 것을 특징으로 하는 매체 스트리머.

청구항5

제1항에 있어서, r을 데이터 스트림의 명목상 데이터 속도라 하고 n을 상기 명목상 데이터 속도로 동시에 출력되는 데이터 스트림들의 최대 개수라 하고 d를 상기 대용량 저장 장치 유닛들중 하나의 유효 출력데이터 속도라 하고 m을 상기 N개 데이터 유닛들 중 최소한 하나라도 저장하고 있는 상기 대용량 저장 장치 유닛 모두로부터 상기 명목상 데이터 속도로 동시에 출력되는 데이터 스트림들의 최대 개수라 할 때 X의 값은 다음 수식 $X = \text{maximum}(r*n/d, r*m/d)$ 에 따라 결정되는 것을 특징으로 하는 매체 스트리머.

청구항6

다수의 N개의 데이터 블록들로서 저장되는 비디오 표현(video presentation)으로서 그 (비디오 표현의) 전체를 표현하는데 시간 T가 요구되고 상기 각 데이터 블록은 상기 비디오 표현의 대략 T/N 기간에 대응하는 데이터를 저장하는 최소한 하나의 비디오 표현에 대한 디지털 표시(digital representation)를 저장하기 위하여 다수의 대용량 저장 장치 유닛으로 구성된 대용량 저장 장치(mass storage)를 포함하는 최소한 하나의 저장 노드(storage node)와, 상기 최소한 하나의 저장 노드의 출력으로부터 비디오 표현의 디지털 표시를 수신하기 위해 상기 최소한 하나의 저장 노드의 출력에 연결된 최소한 하나의 입력 포트(input port)와, 상기 디지털 표시의 소비자에게 하나의 데이터 스트림으로 디지털 표시를 각각 전송하는 다수의 출력 포트들을 각자 구비하는 다수의 통신 노드(communication node)로 구성되며, 상기 디지털 표시의 N개 데이터 블록들은 X개의 스트라이프로 분할되며, 데이터 블록 1, X+1, 2*X+1, ... 등은 상기 X개 스트라이프들 중 첫 번째 스트라이프와 연관되고 데이터 블록 2, X+2, 2*X+2, ... 등은 상기 X개 스트라이프들 중 두 번째 스트라이프와 연관되며, 상기 X개 스트라이프들 각각은 상기 다수의 디스크 데이터 저장 장치 유닛들중 서로 다른 저장 장치 유닛에 저장되며, r을 데이터 스트림의 명목상 데이터 속도라 하고 n을 상기 명목상 데이터 속도로 동시에 출력되는 데이터 스트림들의 최대 개수라 하고 d를 상기 디스크 데이터 저장 장치 유닛들중 하나의 유효 출력 데이터 속도라 하고 m을 상기 N개 데이터 유닛들중 최소한 하나라도 저장하고 있는 상기 디스크 데이터 저장 장치 유닛 모두로부터 상기 명목상 데이터 속도로 동시에 출력되는 데이터 스트림들의 최대 개수라 할 때, X의 값은 다음 수식 $X = \text{maximum}(r*n/d, r*m/d)$ 에 따라 결정되는 것을 특징으로 하는 매체 스트리머.

청구항7

제6항에 있어서, 상기 다수의 디스크 데이터 저장 장치 유닛은 비디오 표현의 디지털 표시에 대해 하나의 사본을 저장하며, 상기 X개 스트라이프들은 다수의 데이터 스트림들이 상기 N개 데이터 블록들 중 동일한(same) 하나의 블록을 동시에 전달할 수 있게 하는 방식으로 판독되는 것을 특징으로 하는 매체 스트리머.

청구항8

제6항에 있어서, 상기 다수의 디스크 데이터 저장 장치 유닛은 비디오 표현의 디지털 표시에 대해 하나의 사본을

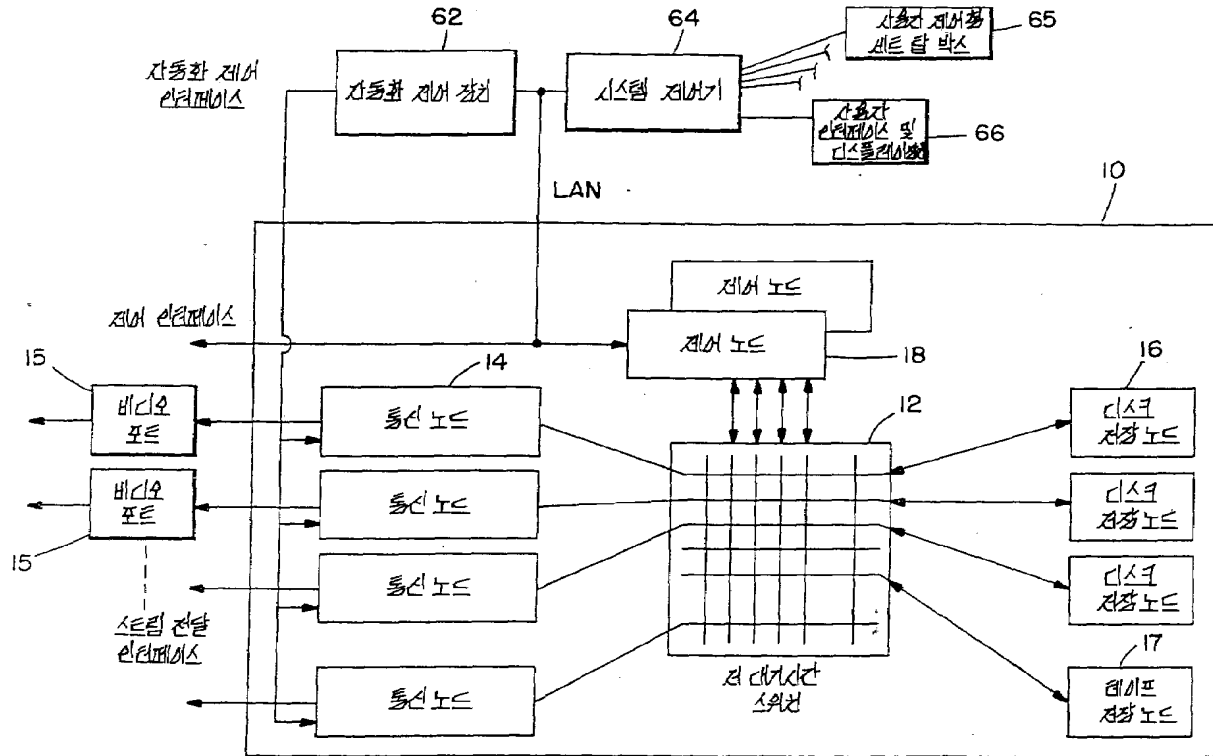
저장하며, 상기 X 개 스트라이프들은 다수의 데이터 스트림들이 상기 N 개 데이터 블록들중 서로 다른(different) 하나의 블록을 동시에 전달할 수 있게 하는 방식으로 판독되는 것을 특징으로 하는 매체 스트리머.

청구항9

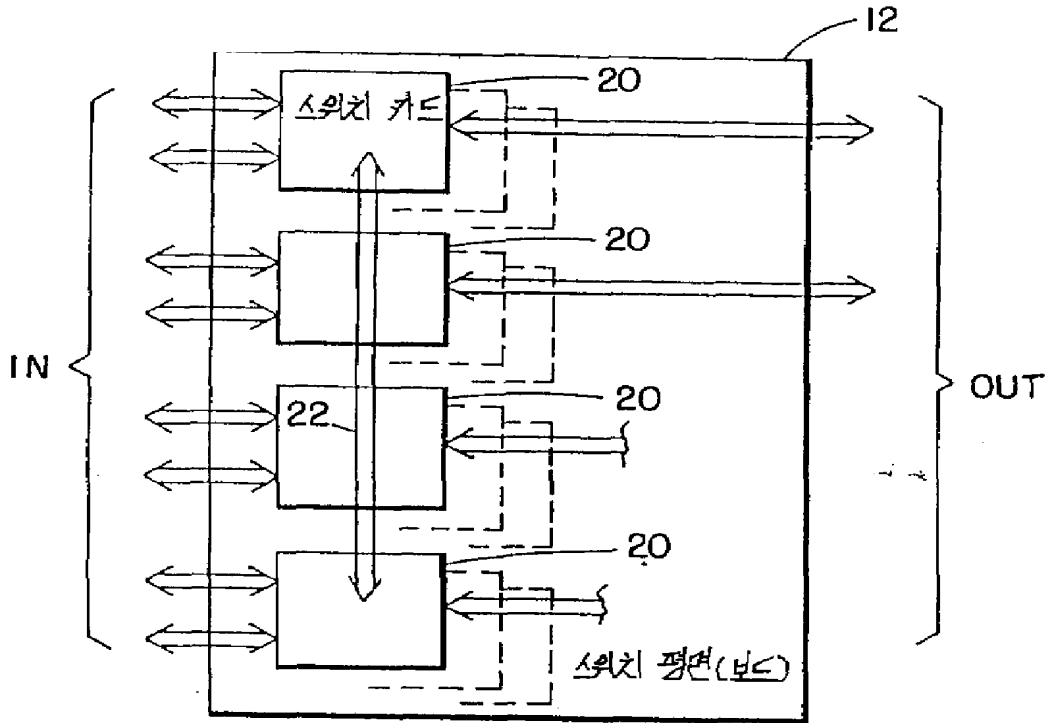
제6항에 있어서, 상기 T/N의 기간은 대략 0.2초 내지 대략 2초의 범위안에 있는 것을 특징으로 하는 매체 스트리머.

도면

도면1

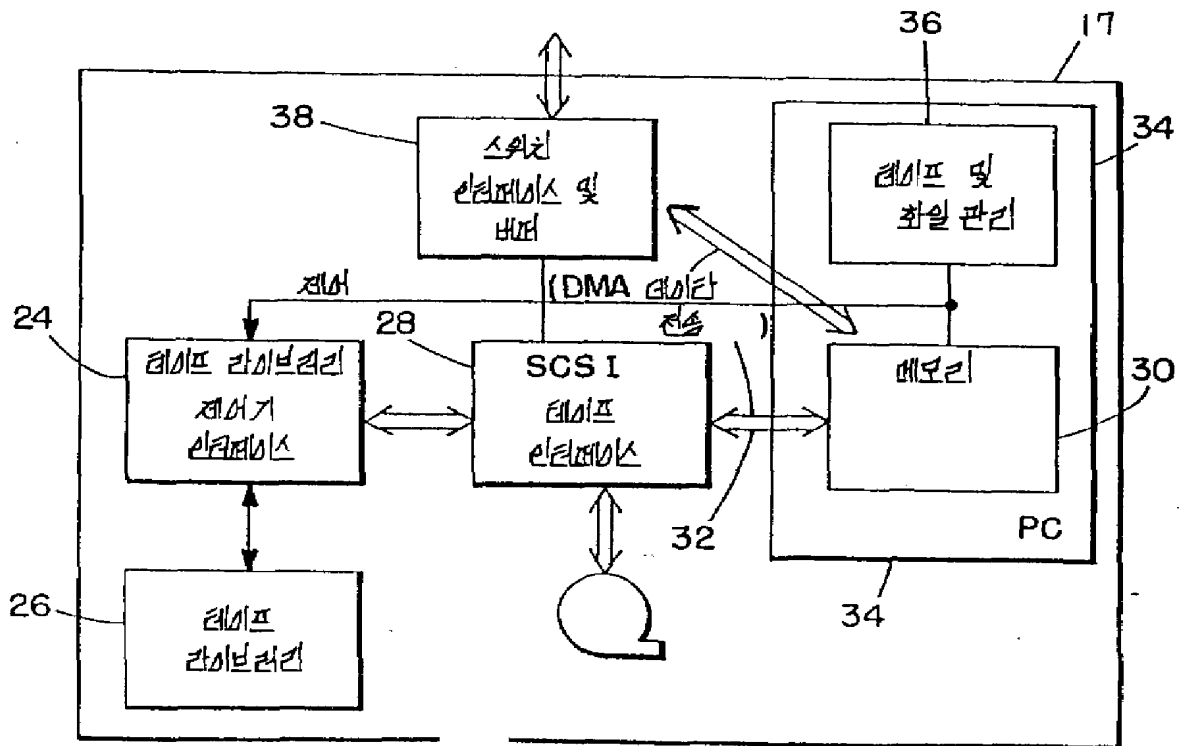


도면1a

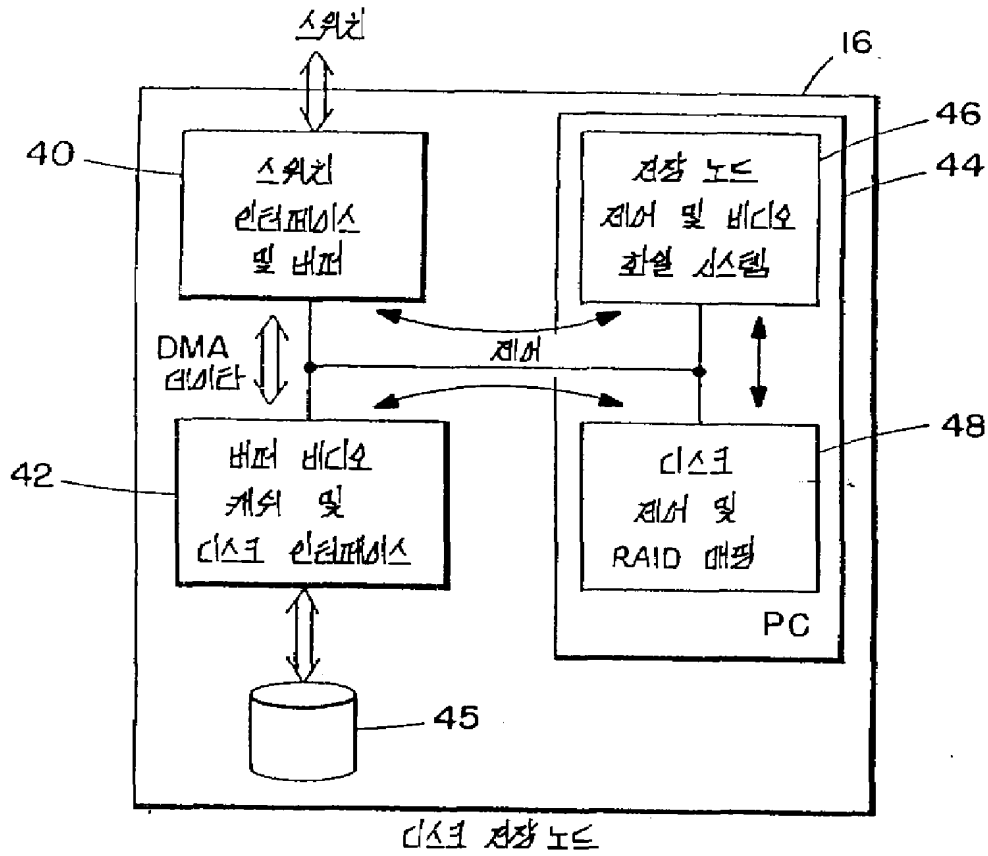


< 저 대기시간을 갖는 크로스바 회선 스위치 >

도면 1b



도면1c



도면1d

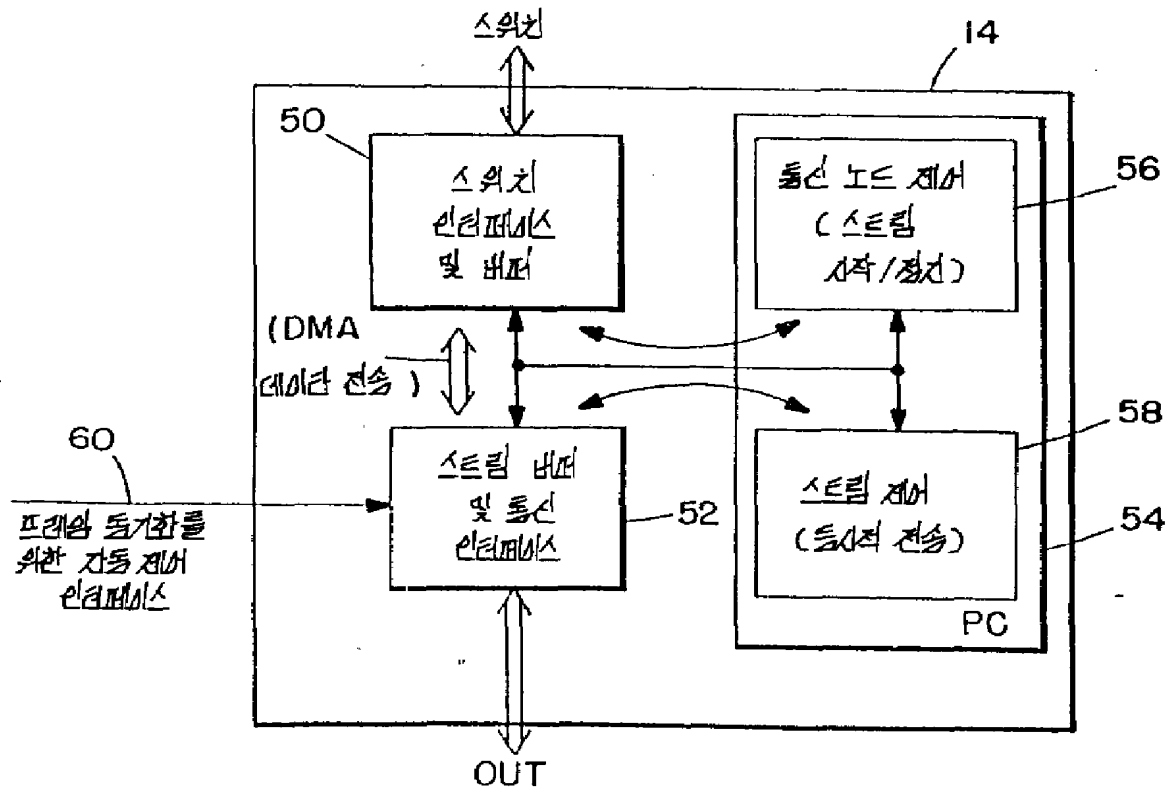


FIG. 2

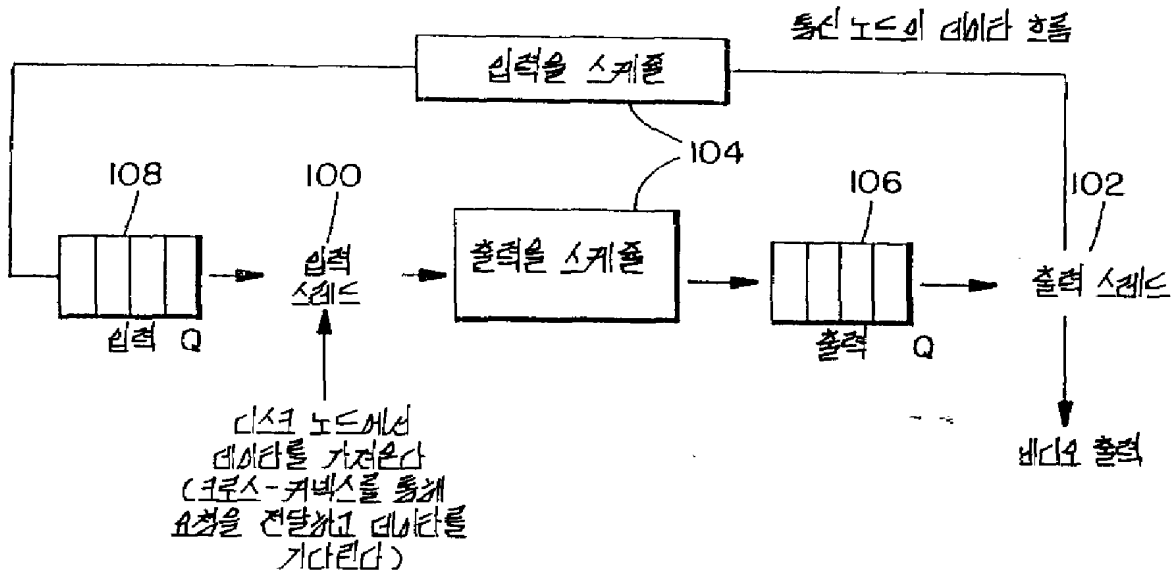
스트림 제어 명령들 (높은 우선순위로 실행)

VS-CONNECT
VS-PLAY
VS-RECORD
VS-SEEK
VS-PAUSE
VS-STOP
VS-DISCONNECT
VS-CONNECT_LIST
VS-PLAY_AT_SIGNAL
VS-RECORD_AT_SIGNAL

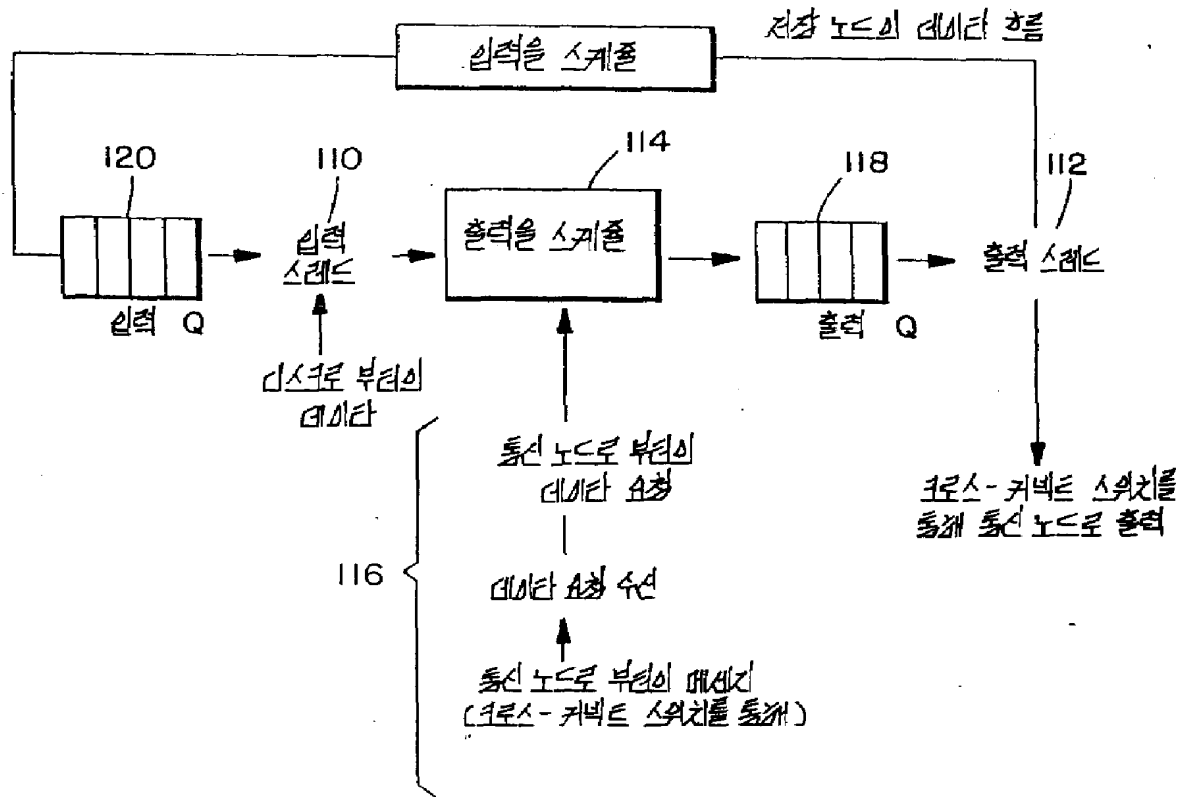
데이터 관리 명령들 (낮은 우선순위로 실행)

VS-CREATE
VS-OPEN
VS-READ
VS-WRITE
VS-GET_POSITION
VS-SET_POSITION
VS-CLOSE
VS-RENAME
VS-DELETE
VS-GET_ATTRIBUTES
VS-GET_NAMES
VS-DUMP
VS-RESTORE
VS-SEND
VS-RECEIVE
VS-RECEIVE_AND_PLAY

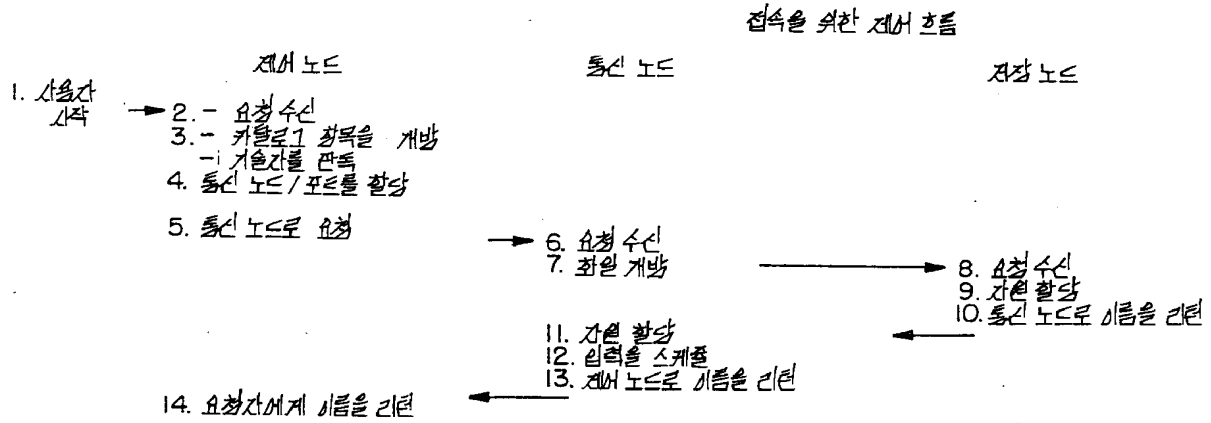
END



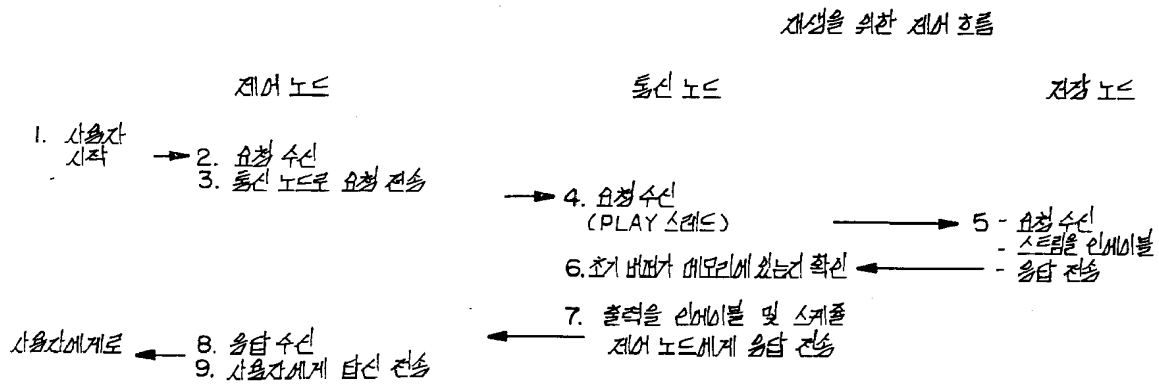
도면 4



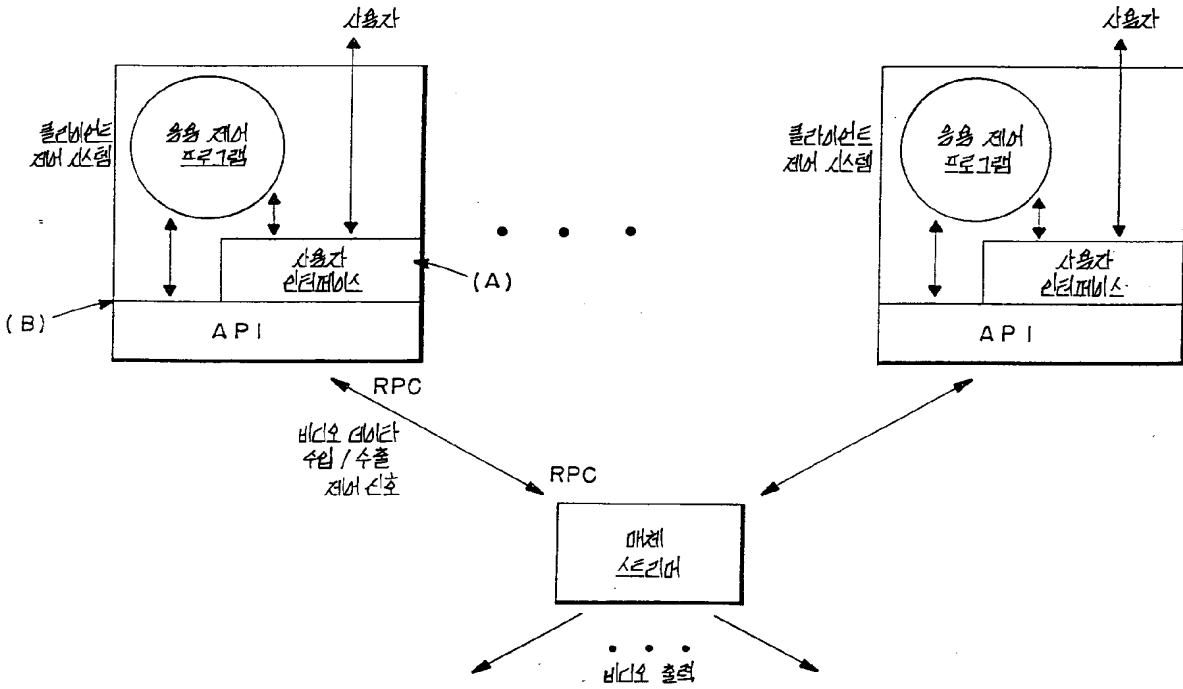
도면 5



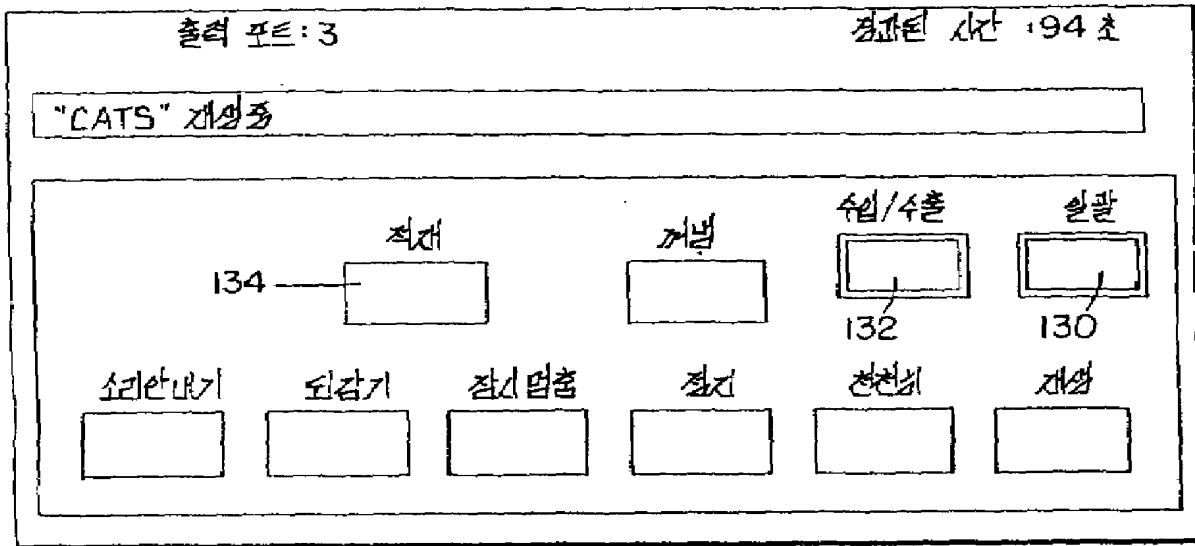
도면 6



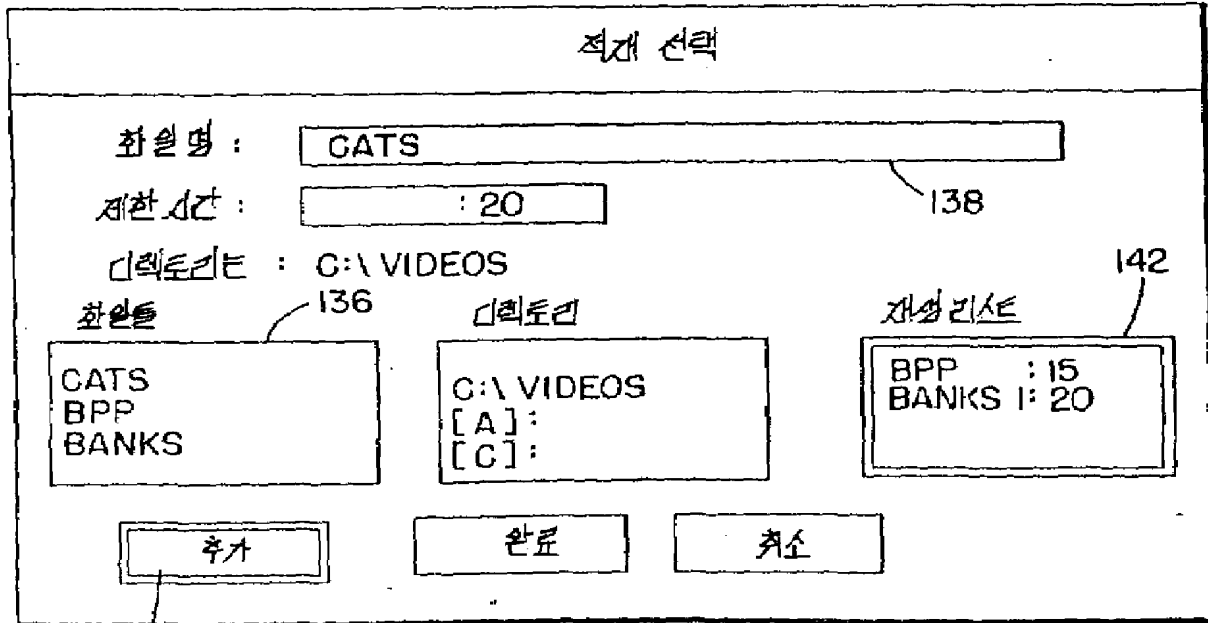
도면 7



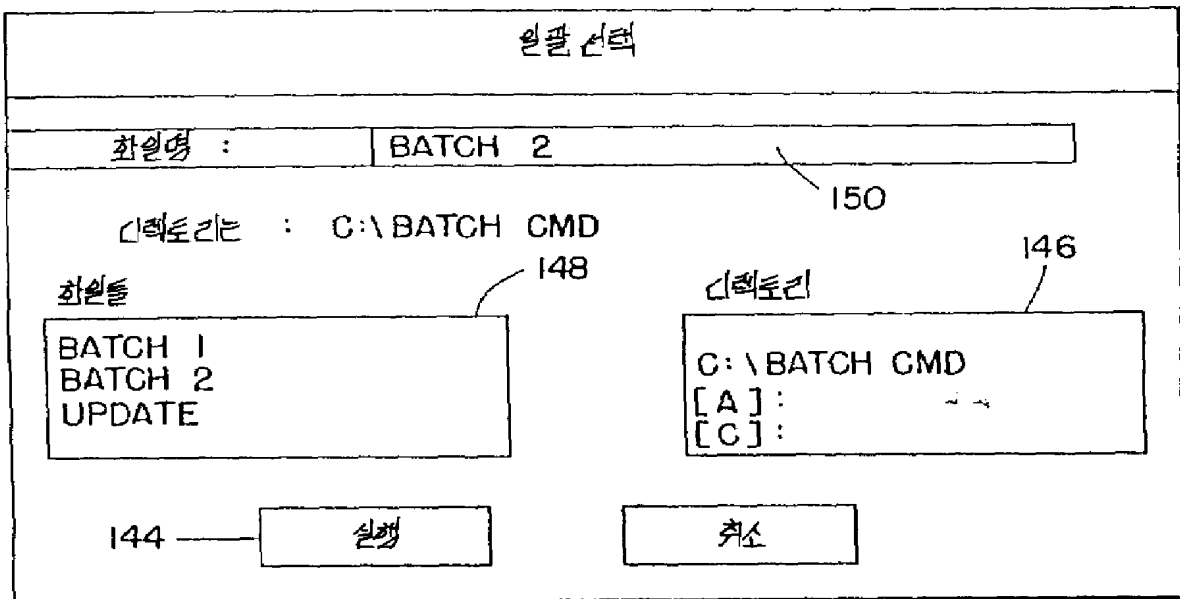
도면 8



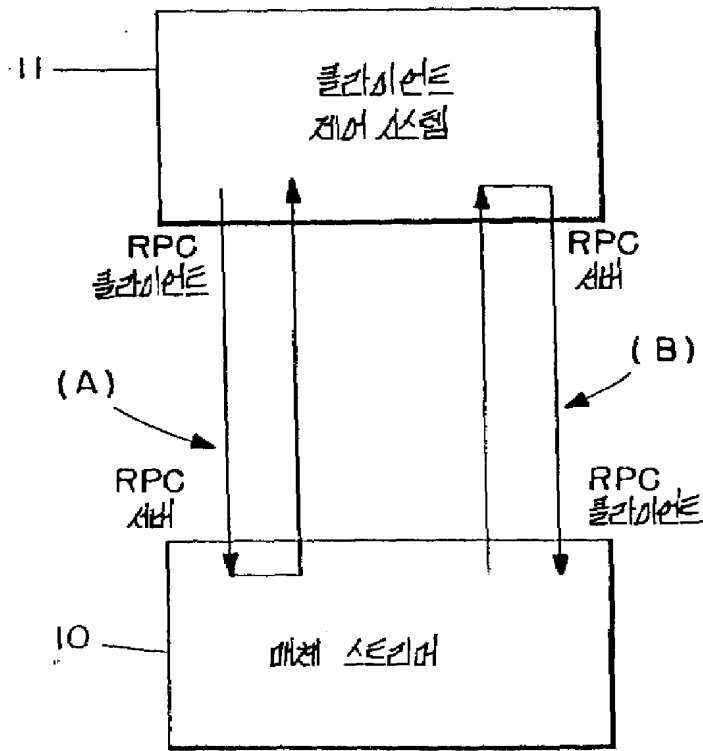
도면 9



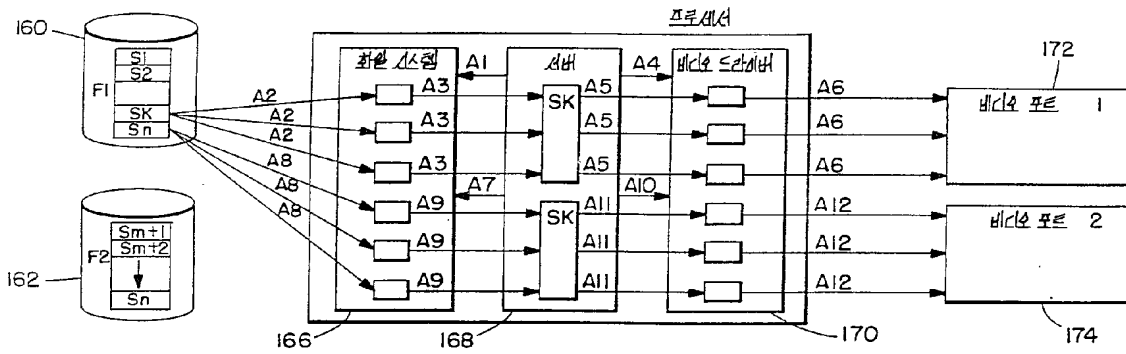
도면10



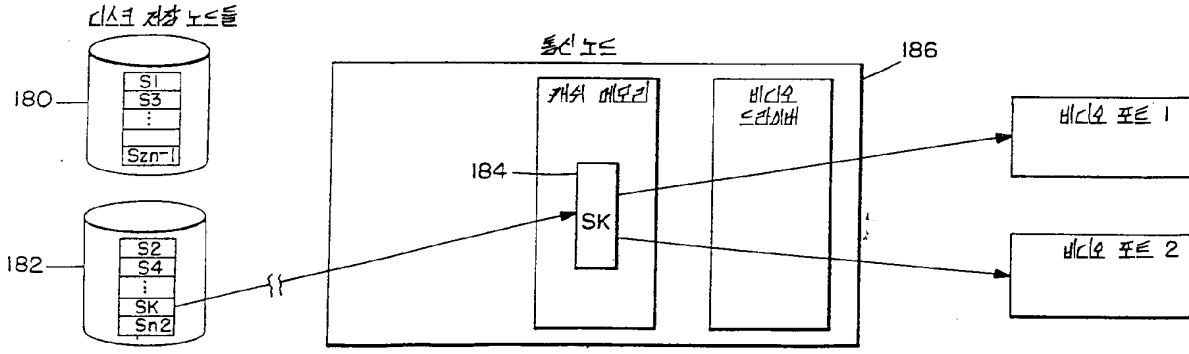
도면11



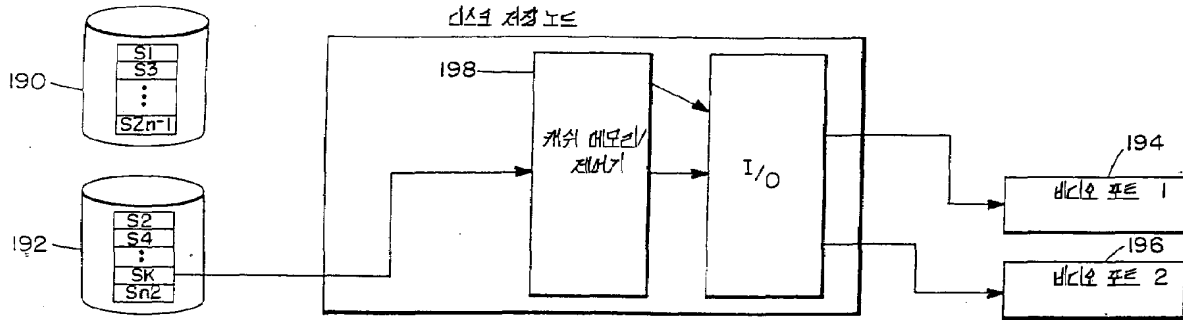
도면 12



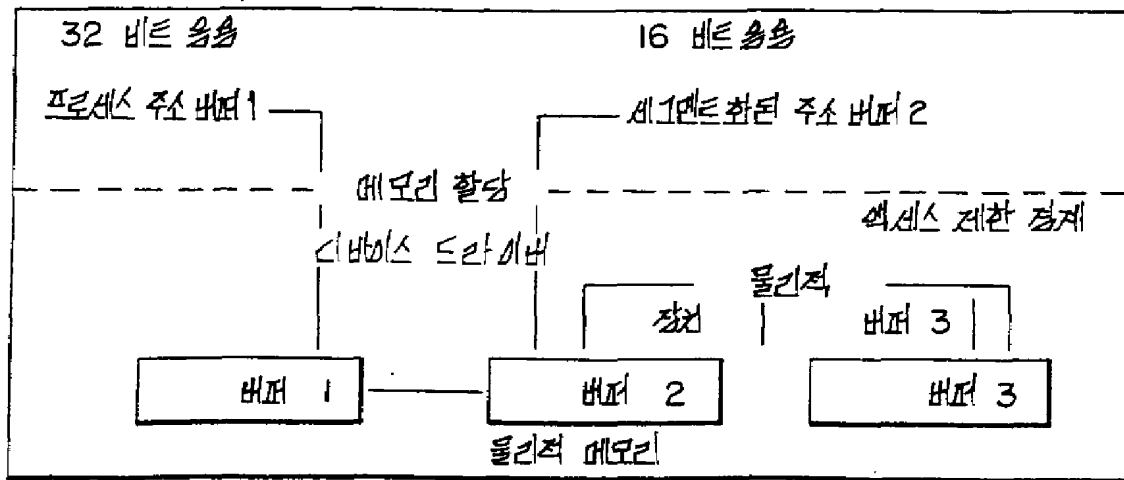
도면 13



도면 13

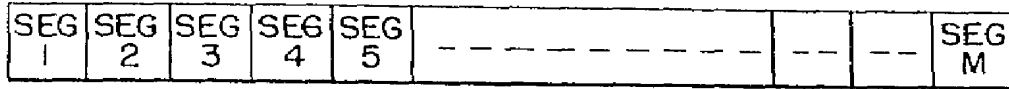


도면 14



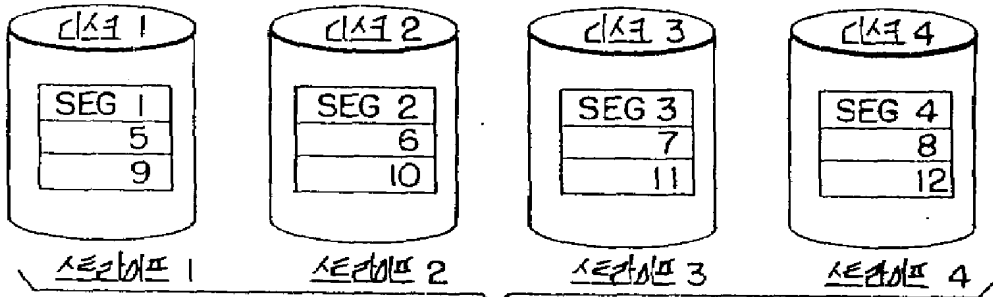
메모리 할당 스킴의 세시적 용용

도면 15



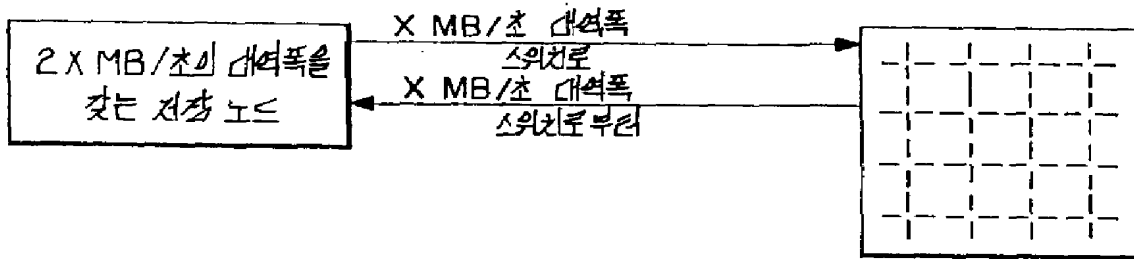
비디오 1개 대한 세그먼트화된 논리적 화질

도면 17



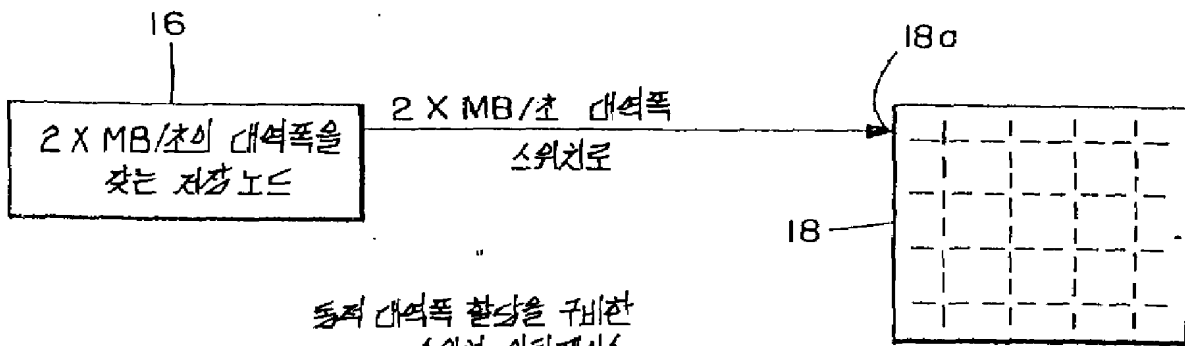
비디오 1

도면 18

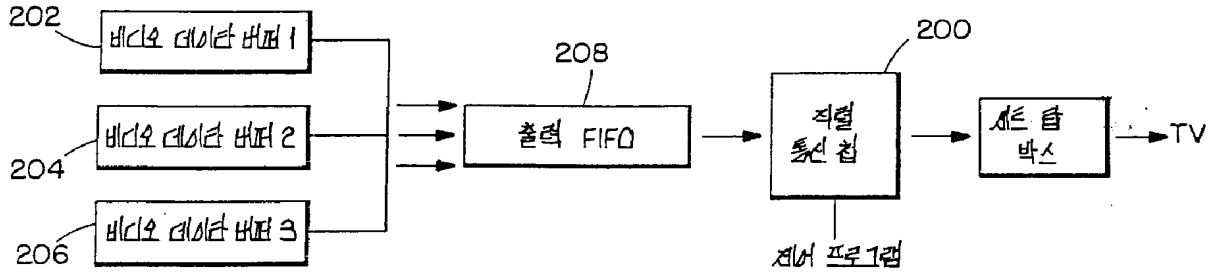


스위치 인터페이스

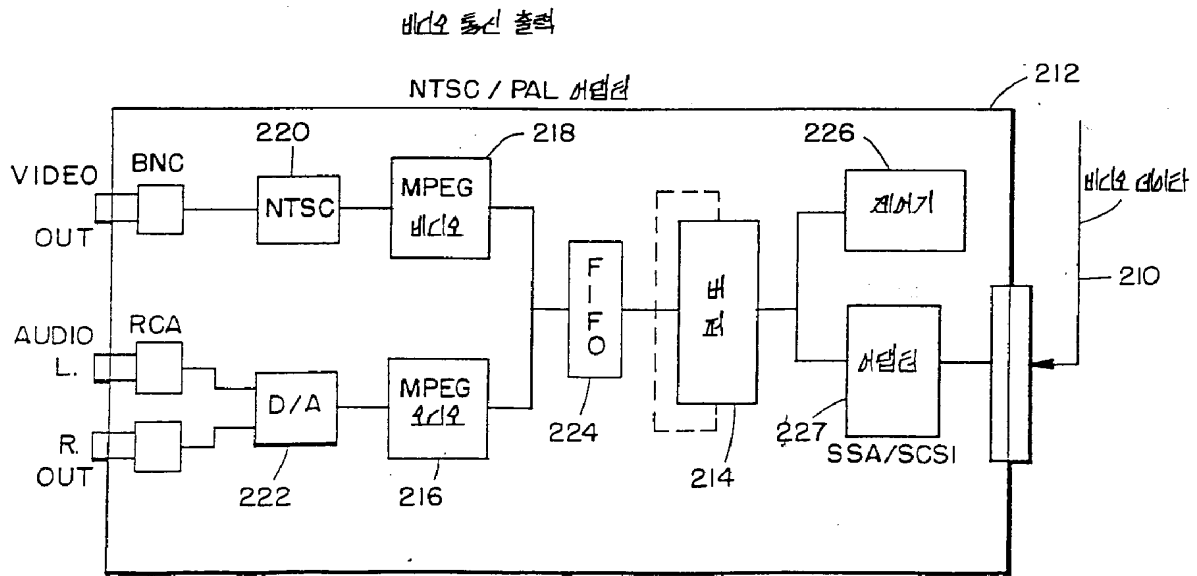
도면 19



도면 20



도면 21



도면 22

